THE UNIVERSITY OF ALBERTA

INTERACTIVE GRAPHICS AND A PLANNING PROBLEM

by

© Gordon Francis Deecker

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

FALL, 1970

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled INTERACTIVE GRAPHICS AND A PLANNING PROBLEM submitted by Gordon Francis Deecker in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

The thesis describes an investigation to consider the potential value of interactive graphics as a planning tool. Attempts by a number of authors to define the planning process have been examined. An interactive graphics implementation has been developed for the Sieve Process, a planning procedure for determining the land-site of a building.

As well as implementing the Sieve Process, certain problems common to map storage and retrieval applications in general have been considered. One common problem is that of choosing the method of digitally representing maps. Three possible representations have been considered. It has been found that no one representation is best for all situations, and that the choice should be very much dependent on what one wants to do. The functions of data manipulation required for the computer implementation of the Sieve Process have been considered to determine the best representation for this case.

A second problem, the function of "windowing", which involves finding the intersection of two regions, has been investigated in detail. Other problems, such as storage economy and hardware limitations, have been considered in somewhat less detail.

The author believes that the interactive system developed is a practical first step towards a system which could deal with a wide range of planning problems.

ACKNOWLEDGEMENTS

I express my appreciation to Professor J. P. Penny for his advice, guidance, and patience with me throughout the duration of this research. Special thanks are extended to Larry Loh for his discussions on the planning process, to Curt Enerson for aiding with the programming, to Howard Pangowish for the many illustrations in this thesis, and to my wife Dorothy for the typing and proofreading.

I wish to thank the Department of Computing Science for the financial assistance and facilities which have made this project possible.

TABLE OF CONTENTS

ILLUSTRATIONS

CHAPTER I

INTRODUCTION

This thesis describes an investigation into the potential value of interactive graphics for campus planning. The use of computers for map storage and retrieval problems in general has so far been relatively small. Many of the problems to be solved for applications which would involve both interactive graphics and map data are not at all well understood.

The planning process, as applied to site choice and building construction, involves decision-making based on map and numerical data manipulation. For example, to determine the land site of a new university structure, the planner must coordinate maps of present building locations, zoning regulations, campus parking facilities and road access routes, and so on, with cost analysis data, expected student usage and financial budgets. Many intermediate results obtained in the course of the investigation not only affect the final outcome, but also influence the studies made to obtain the plan finally chosen as the best alternative.

Interactive graphics could, in theory, give the user the ability to see the results graphically at intermediate points and describe changes to the program based on the previous results. The question thus arises: To what extent might interactive graphics have a practical value in this type of planning process?

The planning process is complex. Many articles have been written in an attempt to describe the methodology of the process. The author has examined five approaches to planning: the work of Aguilar (1968, 1969); the work of Myer and Krauss (Berkeley, 1968); the Duke University study (1967); Dror's (1961) theoretical approach; and the systematic approach of Catanese and Steiss (1968). It can be seen that there are basic steps common to each method.

A sixth and little documented approach - the Sieve Process - is deemed by the author to be particularly amenable to the use of interactive graphics. The Sieve Process is characterized by the use of grey scale map drawings to represent relevant planning systems of the area under study (for example, pedestrian paths, campus access routes, or open spaces). These maps may be overlaid to illustrate graphically the various areas where combinations of the required resources are available for the new structure. The areas indicate potential development sites. The planner may then sketch, on the maps representing each of the planning systems, his appraisal of the impact of the new building at any particular location. In this way, possible flaws, such as traffic congestion or lack of sufficient pedestrian access, may be seen. The appraisal, based on such factors as numerical data supplied in the terms of reference given to the planner, or the cost of supplies, may involve computer processing.

Using the University of Alberta campus as an example, the author has implemented the Sieve Process for an interactive graphics terminal with a limited number of actions available to the planner. A principal objective was to discover the major problems to be solved, and to study in detail some of these problems which are common to map storage and retrieval applications.

Maps are made up of what Freeman (1961a) has called "arbitrary lines", that is, lines not necessarily subject to mathematical formulation. Such lines must be approximated by strings of (usually small) vectors for storage in a digital computer. The most obvious digital representation is the storage of coordinates of the end points. We shall call this XY coordinate data. At least two other methods of encoding have been discussed in the literature: chain encoding (Freeman, 1961a, 1961b) and skeleton encoding (Rosenfeld and Pfaltz, 1966; Pfaltz and Rosenfeld, 1967).

The three encoding methods have been studied to determine their relative merits. The study has shown that no one method is best overall. An important conclusion is that the choice of method is, or should be, very much dependent on which functions of data manipulation are required in the application considered. For example, skeleton encoded data is best if the intersection of two arbitrary regions is required. However, it is very unsuitable if the perimeter of a region is required.

In most map storage and retrieval applications, the graphical display could be used as a "window" for viewing stored maps. Parker (1967) gives an illustration of the CRT used as a "zoom lens" viewing, selectively, portions of a map drawn on a very large imaginary surface. The map section viewed on the screen is the intersection of the field of view with the encoded map. The user might view many of these sections during the course of one session at the terminal. While Sutherland (1968) has described hardware (the "Clipping Divider") for determining the map section to be viewed, such hardware is uncommon. Therefore, algorithms for finding the intersection of two regions must be considered in some detail. Since this operation is something that could be done repeatedly, the time required for processing data stored in each form could be an important consideration when choosing a method of encoding.

Other problems encountered while implementing the Sieve Process are also discussed. Functions which may be used repeatedly are considered from the point of view of minimizing processing time requirements. Limitations of the computer system used are also discussed.

The Sieve Process is intended for use in solving land-site selection problems. The Sieve Process may be used at least as a partial base for a more extensive project including also the design and construction of a building. The Duke University study shows one potential path in the university environment. Urban planning could also be

feasible in a similar way.

These extensions are possible, in part, because the map form is a basic tool not only for the Sieve Process but also for many other planning disciplines. That the map form is a basic tool means also that problems encountered when considering any one planning system will often be common to many planning systems. Thus, the definition of these problems and the finding of possible solutions take on added importance.

CHAPTER II

THE PLANNING PROCESS

2.1 INTRODUCTION

Use of computers has greatly changed the mode of operation of many industries. However, it is only recently that computing scientists have entered the realm of the planners and architects. Some examples of a changing attitude can be seen. MIT (Berkeley, 1968) has organized a Center for Building Research. A group of planners at Duke University (1967) is studying "computer-aided techniques which will help higher educational institutions deal with academic, architectural, and financial aspects of planning campus facilities." Aguilar (1967a, 1967b, 1968, 1969) of Louisiana State University is studying the use of Linear Programming in relation to architectural planning.

It is apparent from reading the literature that a rigorous definition of planning is more of an ideal than a reality. Town planners, civic planners, urban planners, architectural planners, each seem to have a notion of what planning is all about. There is, however, some consensus on the methodology of planning. In this section we will consider the work of the people mentioned above, and two outlines of the planning process in general, in order to obtain some ideas on the basic formulation of the process.

2.2 THE PLANNING PROCESS: AN OUTLINE

Catanese and Steiss (1968) have listed the basic components of the planning process as follows:

> "(1) Identification and definition of problems and their interrelationships;
> (2) determination of goals and objectives associated with each problem situation and the problems in totality;
> (3) appraisal of existing policies and procedures designed to achieve goals and objectives;
> (4) formulation of available alternatives to reach agreed goals and objectives;
> (5) evaluation of alternatives;
>   a) identification of by-products and side-effects;
>   b) determination of approximate benefits and costs associated with each alternative;
> (6) recommendation of appropriate alternative."

Dror (1961) in his study on planning delineates three phases, and with each phase are associated several tasks. These phases and tasks are as follows:

(a) Pre-planning -
- (1) Preliminary determination of planning objectives.
- (2) Setting down of terms of reference.

(b) Planning -
- (1) Translation of planning objectives into weighted operational goals.
- (2) Collection of information.
- (3) Search for main alternatives.
- (4) Relative evaluation of the main alternatives and identification of the optimal one.
- (5) Re-examination of the optimal alternative with the aid of additional information and search.
- (6) Translation of the optimal alternatives into a set of proposals for action in the future.

(c) Post-planning -
- (1) Plan approval.
- (2) Plan execution.
- (3) Examination of results in the light of the plan.
- (4) Feedback.

From the above outlines, one is able to obtain some

appreciation of the manner in which a planner goes about his task.

One important point which is latent in the above outline is the subjective nature in many instances of the decisions required.

2.3 THREE APPROACHES TO THE PROBLEM

2.3.1 STRUCTURAL OPTIMIZATION

Aguilar (1967b) points out that the design of a building involves four steps:

"(1) PROBLEM IDENTIFICATION AND DEFINITION OF GOALS, requiring complete understanding, not only of the structure of the situation under consideration, but also of the objectives to be attained by the design solution.

(2) GENERATION OF ALTERNATIVES for accomplishing the preset objectives. This component is highly dependent upon the designer's imagination and upon his scientific-creative ability.

(3) DEFINITION OF AN OBJECTIVE FUNCTION to evaluate the worth of each alternative in fulfilling the problem's objectives.

(4) OPTIMIZATION OF THE OBJECTIVE FUNCTION, leading to the most effective system's configuration, i.e., the optimal design solution."

Once several alternatives are identified they are evaluated in terms of the Objective Function - generally a mathematical function expressing cost and/or profits. In this manner, one may decide on the optimal structural configuration, i.e., the design which best satisfies the objectives and goals as previously set out. It is interesting to note that, once the optimal solution for this criterion has been determined, the "cost" of aesthetic considerations

can be evaluated in terms of the additional expense over and above that of the optimal solution.

2.3.2 DYNAMIC CONSIDERATION

Myer and Krauss (Berkeley, 1968) have considered, as an example, the design of a nursery school. They say that they did not proceed "by analyzing carefully a set of goal statements, setting up a different geometrical solution for each, and then proceeding to resolve the relationship among these units."

In point of fact, at a quite late stage, the clients changed their mind as to the definition of the system required. Myer and Krauss found that some requirements, such as "a special location for story-telling", were too dependent on aesthetic factors to determine an objective function. It was found that the process was "dynamic", there being a continuous cycling of form and criteria until all conflicts were resolved.

To summarize their conclusions: "the final form is the direct result of the order in which a designer chooses to consider variables". Any computer approach must allow this freedom.

2.3.3 PLANNING IN A UNIVERSITY ENVIRONMENT

A group at Duke University (1967) is investigating many aspects of campus planning. A room classification scheme has been developed. Room and equipment inventories have been instituted for use in planning for production resource allocation. Activities on campus have been

classified and studied to determine densities of facility usage and the affinities among the various facilities. Site conditions are recorded as well as monies available.

The various phases of the procedure are shown in Figure 2.1. Basically, the report of their studies proposes a method whereby a total inventory of the university resources may be recorded, and a projection of future expansion may be obtained from current trends.

2.4 SUMMARY

It is important to note that, independent of the method or application involved, each of the reports discussed agrees on the necessary steps involved in determining a solution to the design or planning problems. These steps are as follows:

(1) Determination of Planning Objectives.

(2) Determination of Various Alternatives.

(3) Evaluation of the Various Alternatives.

(4) Optimization of Evaluation Techniques to Determine the Optimal Alternative.

2.5 CONCLUSION

In the reports by Aguilar and by Myer and Krauss we see three important points that must be considered in any procedure for solving planning problems.

(1) There are definite considerations such as cost, weight, strain, etc., which must be considered in evaluating any alternative choice. These

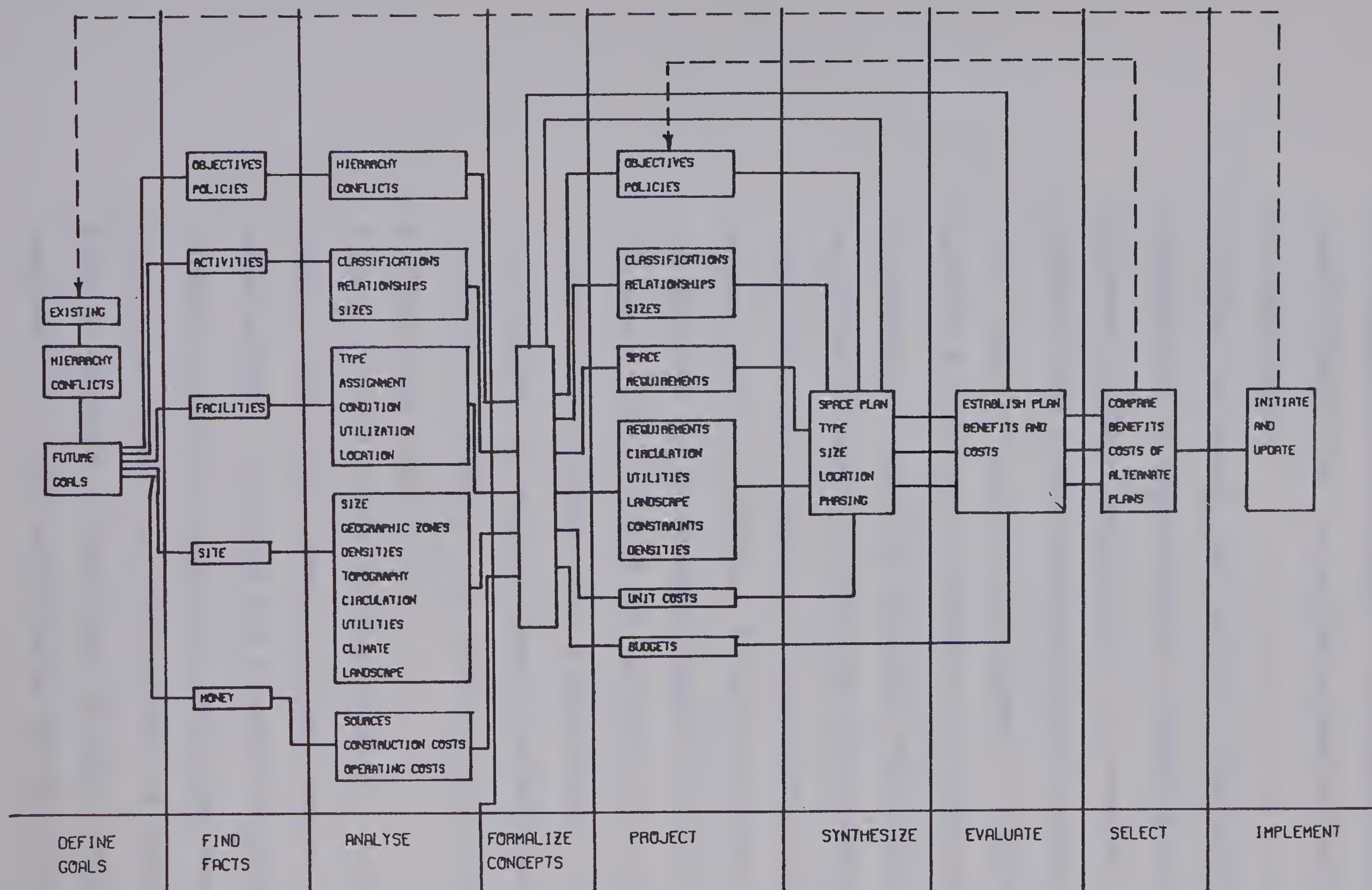


Figure 2.1 Duke University Campus Planning Scheme

factors are too complex to permit the designer to evaluate them manually. A computer must be used if more than one or two alternatives are to be considered.

(2) The plan or design that results from applying the process is traditionally viewed by most planners as the "progeny" of its creators, endowed with their personality and perspective. This view, a result of the number of subjective decisions made by the planners in the course of the process, forms the greatest stumbling block to the acceptance of computers by planners. Given the same criteria, five planners would determine five different solutions to the problem. The results of any design competition bear witness to this. What is required is a method whereby this singularity is maintained, while the drudgery of menial tasks is eliminated. Interactive graphics, in theory, could give the planner this capability.

(3) Beauty is in the eye of the beholder. Choices based on aesthetic considerations cannot be computerized. In fact the planner in many cases cannot explicitly state that which he considers beautiful. Thus, all evaluations by computer will be lacking in that they do not consider aesthetics. This shortcoming may be partially

overcome by use of the graphical display, where the designer can see the form he is creating. In this manner the "dynamic process" required by Myer and Krauss is given new meaning and added emphasis. The designer has control over his creation, and a computer at his bidding to perform complex computations.

In the next section we shall consider a specific planning problem - that of land-site selection. The method used is the Sieve Process. The method is described in relation to the four steps above. A possible implementation of it, or at least that portion which Dror calls the planning phase, is described. It is left to the planners to decide how well the implementation will allow the singularity they desire and a facility to consider aesthetics.

CHAPTER III

THE SIEVE PROCESS

3.1 INTRODUCTION

It was shown in the previous section that any problem-solving procedure in planning should be a union of four distinct steps: determine planning objectives; determine possible alternatives; evaluate the alternatives with respect to the objectives; optimize evaluation methods in order to determine the best alternative. In this section, a specific planning process is considered - the Sieve Process, a procedure for determining the site of a new building. After outlining the process, the author will show how the process conforms to the four step formula. While the process is applicable to a wide range of situations, the inventory, that is, data required by the planner, is completely dependent upon both the type of structure to be built and the terms of reference. The inventory given here is that which is required for determining the site of a building in a university environment.

3.2 METHODOLOGY

3.2.1 TABULATION

The major reference for the campus planner is a BASE MAP of the university, which outlines all existing buildings, roads, and major tree lines.

The first step is to consider the resources. The name used by the planner for these resources is "operating systems".

For each of these systems the planner draws a separate map. Each map is composed of regions coloured white or black. Regions suitable for a new building with respect to the operating system are white, while regions unsuitable are black. For example, on a system map for buildings, the site of existing buildings must be black, and a vacant lot would be white. Typical operating systems maps and what they illustrate are:

(1) TOPOGRAPHY- 10 foot contour intervals over the campus with, possibly, dark regions showing poor soil conditions.

(2) EXISTING STRUCTURES- sites of present buildings along with a 20 foot peripheral area to be left open (unless some physical link between buildings is desired) would be black.

(3) VISUAL CHARACTER- the visually perceived "units" of the campus and their zones of influence, that is, a differentiation between residential, academic and social areas. Depending on the type of building to be built one or two of the areas would be black. For example, a law school building should be in the academic area.

(4) HISTORICAL AND SOCIAL CHARACTER- circles would indicate gathering points, such as a quadrangle. In this case, black areas would indicate those areas which should be left untouched.

(5) UTILITIES- major utility lines and service access roads - areas beyond, say, 150 feet from these lines might be black since building so far from existing utilities could be uneconomic.

(6) PARKING- parking areas and access roads would be black.

(7) PEDESTRIAN EASEMENT- black regions would delineate major pedestrian links which should, as far as possible, be retained and perhaps improved.

(8) MAJOR TREE PLANTINGS- black areas would indicate major planted areas that contain mature trees which ought to be retained wherever possible.

Figure 3.1 shows the PEDESTRIAN EASEMENT operating system map of the University of Alberta.

3.2.2 HOW THE SIEVE WORKS

The planner draws these maps on transparent plastic. When the maps are overlayed resultant dark zones indicate the presence of the resources of the campus. The light (unshaded) zones indicate unused portions of the campus. For any particular problem the planner draws up a list of objectives using the terms of reference stated by the administration as a guide. These objectives may be satisfied by considering various operating systems. Figure 3.2 lists one possible set of interactions. The lines indicate which maps (systems) must be considered when the operational goal is considered. The planner may determine which operating systems (generally two or three) are most important in the context of his

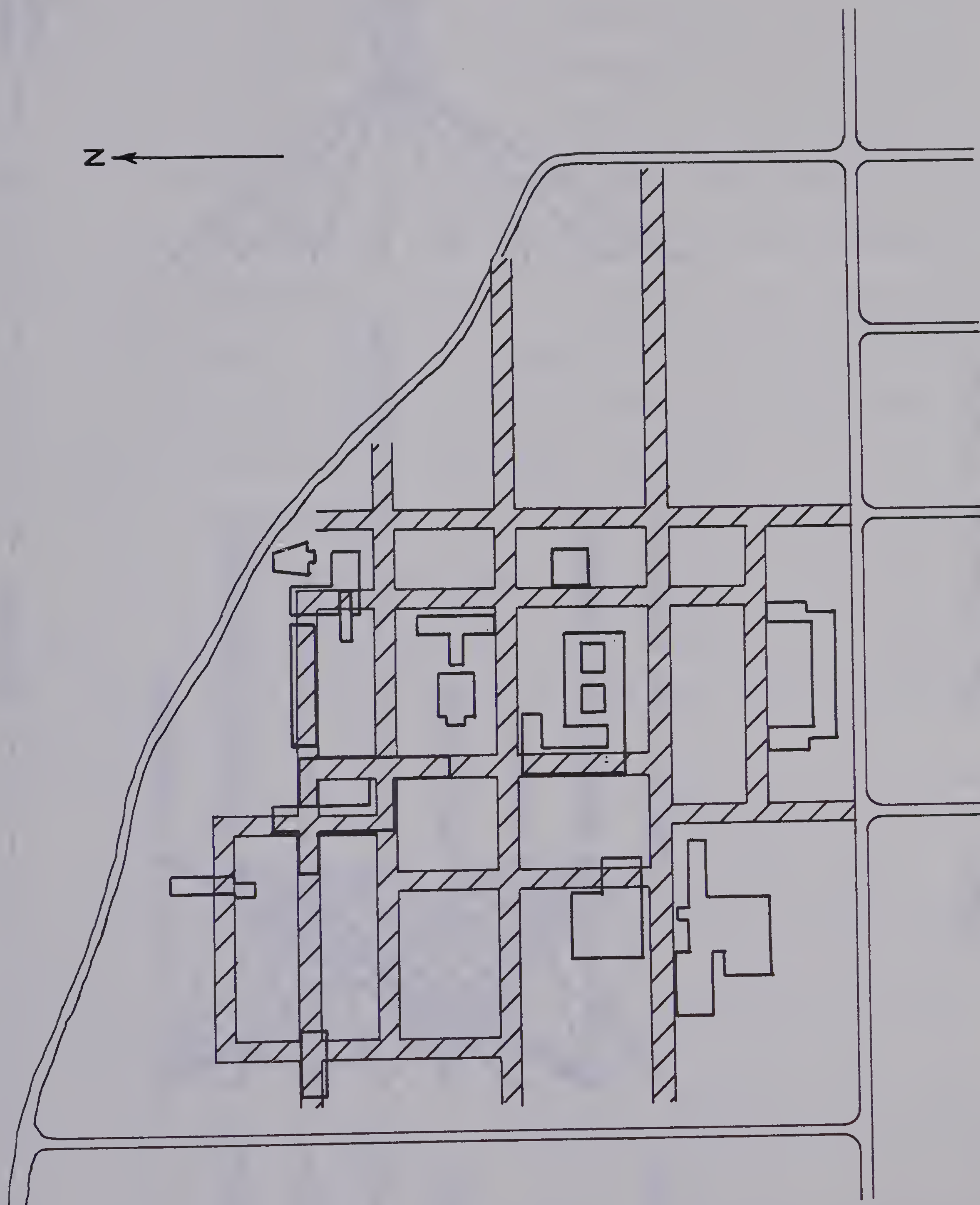


Figure 3.1 Pedestrian Easement

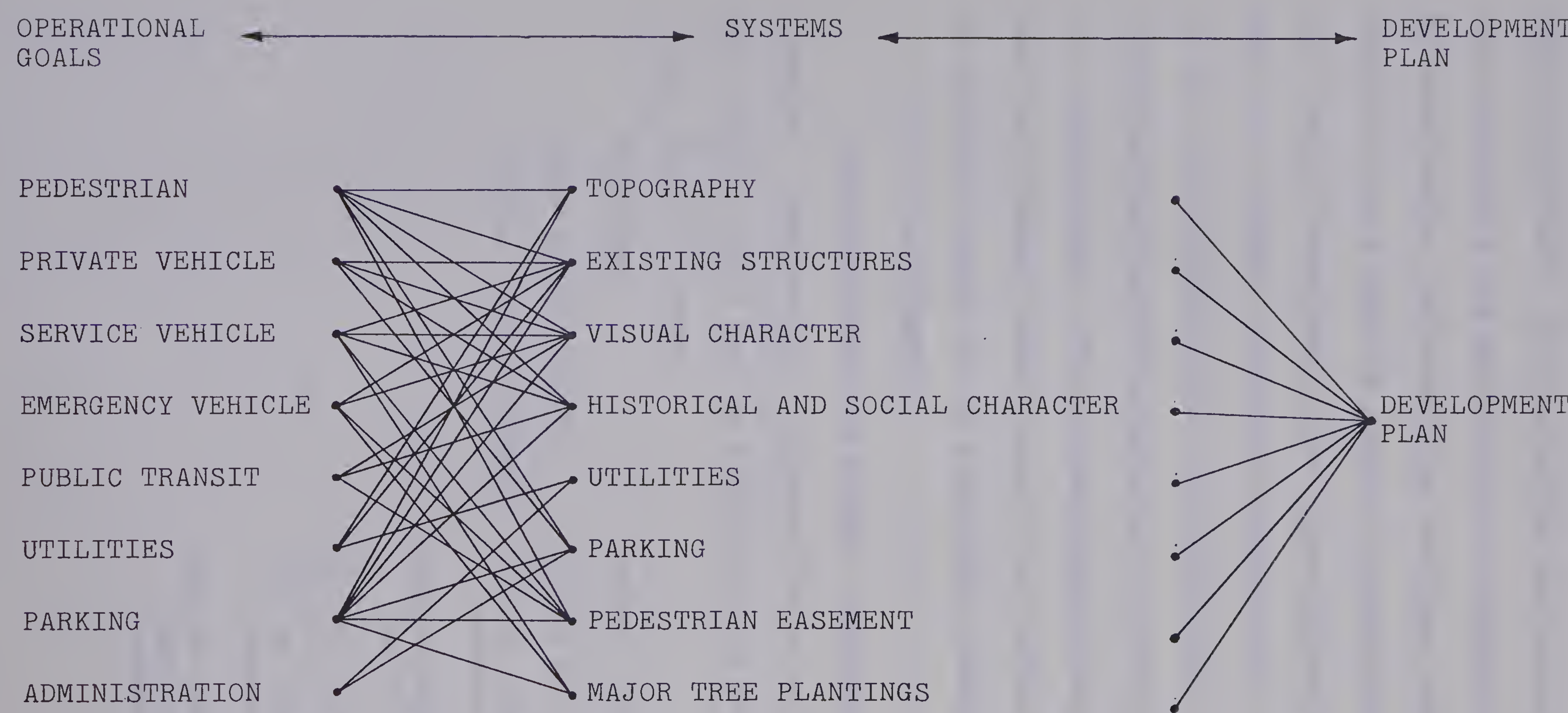


Figure 3.2 The Planning Process

current problem. By overlaying the resource maps for these systems first, the main areas of interest can be located. By considering the remaining overlays, these areas may be refined into a list of several alternative site locations. By viewing these land sites superimposed on the resource maps, the planner is able to narrow his investigation to one or two site alternatives. At this point a more thorough investigation may be considered to determine the best location for the building. This investigation may involve other items such as climate, noise nodes, etc., which are too variable to consider except for a particular location.

3.2.3 AN EXAMPLE

The process may be better understood if we take an example. Considering a Student Union Building, one can see why certain sites may be rejected, and how this is determined in the Sieve Process.

Building:	Student Union Building.
Space Requirements:	90-100,000 square feet.
External Requirements:	(1) be located along a major student pedestrian artery.
	(2) be adjacent to a major parking facility.
	(3) be accessible for service.
	(4) be close to existing student activities.
	(5) be accessible to faculty and visitors as well as students.

Some possible criteria for rejection are:

(1) A location may be rejected because of poor ground soil. This would be determined by viewing the location superimposed on the TOPOGRAPHY map.

(2) A location may be rejected because it is not close enough to parking facilities. This would be determined by viewing the location superimposed on the PARKING map.

(3) A location may be rejected because it is not accessible enough for the frequent servicing required for the dining facilities. This would be determined by viewing the location superimposed on the UTILITIES map.

3.3 THE FOUR PLANNING STEPS

The Sieve Process may be summarized by considering it in relation to the four basic planning steps.

(1) Determination of Planning Objectives.

The planner is presented with terms of reference, that is the goals, objectives and requirements as set down by the administration. From these a list of priorities can be established.

(2) Determination of Various Alternatives.

By sieving through the resource maps the planner is able to determine several alternative locations for the structure required. These are probably chosen after considering only those regions which satisfy the three or four most important requirements.

(3) Evaluation of the Various Alternatives.

These areas are then evaluated by the planner with respect to all the requirements in order to find what appears to be the best alternative.

(4) Optimization of Evaluation Techniques to Determine the Optimal Alternative.

Once the study is limited to a specific area, more thorough documentation of the resources in that area may be considered. In many cases it will result in the area being rejected and another alternative from those found earlier being considered. The final result is that one site is selected.

3.4 STANDARDS

From the preceding discussion, a number of requirements are evident:

(1) The overlay maps must be usable both independently and in any desired combination.

(2) The scale of presentation must be variable to enable the planner to look at parts of the map in greater detail when necessary.

(3) The maps should be graphically legible at all levels of overlay and in any combination of overlays.

3.5 SUMMARY

We have seen how a planner using the Sieve Process at present approaches the problem of campus planning. Transparencies are drawn outlining the resources available. When overlayed for consideration together, an indication is given of areas where a multiplicity of resources are available. These areas are then considered in greater detail, until one area is chosen as the best site.

Let us now consider how the same process may be done using a graphical display terminal.

CHAPTER IV

THE SIEVE PROCESS - A COMPUTER APPROACH

4.1 INTRODUCTION

It was shown how the planner currently uses the Sieve Process to determine the site for a new building in the university. In this section one possible method of solving the process using a graphical display terminal is considered. Since the author is a computer scientist, rather than a planner, there will be an emphasis on isolating the problems associated with implementing the process - particularly where any problem is seen as one likely to be encountered in implementation of several different planning methods.

The solution described is one that has been at least partly implemented on the computer system currently available at the University of Alberta, that is, an IBM 360/67 computer and a Control Data graphical display (GRID). Dror's outline of the planning phase is used as a guide in the discussion following.

4.2 THE PLANNING PHASE

The six tasks of the planning phase would, when a computer is used, be accomplished as follows:

(1) Determination of the Operational Goals - They would be specified as outlined in the previous section.

(2) The Collection of an Inventory - In the Sieve Process the inventory consists of resource maps

and a base map. These would be maintained on file (e.g. in digitized form on magnetic tapes) as in the Duke University study, and continuously updated. Alternatively the user might define his resource maps by drawing them on the graphical display.

(3) Search for the Main Alternatives - This is accomplished by overlaying the resource maps and viewing the resultant display. Using the CRT the planner is able to see each resource map independently as well as any combination of overlays of these maps. Thus he can determine from viewing the display those areas which contain some or all of the resources required.

(4) Relative Evaluation and Identification of the Optimal Alternative - The regions found in Step 3 above may be considered in relation to the individual resources and a value assigned which depends on how well they fulfil the requirements drawn up in Step 1. This evaluation may be done visually by the planner, or by computation in the computer. In this manner one region would be chosen for more detailed study.

(5) Re-examination of the Optimal Alternative with the Aid of Additional Information and Search - The region chosen in Step 4 would be viewed on a larger scale, thus allowing a more detailed

study of the area. If the area is rejected another would be chosen from the set found in Step 3 above.

(6) Translation of the Optimal Alternative into a Set of Proposals - Once the optimal site has been identified proposals may be drawn up to utilize the site to best advantage.

4.3 REQUIREMENTS

From the foregoing discussion it is evident that there are several major requirements in order to implement the system. Among the most important are:

(1) A method of encoding maps.

(2) A method of map storage and retrieval.

(3) The following functions for manipulation of the encoded maps: scaling, windowing, editing (deleting from or adding to a map), overlaying two maps to obtain the outline of the regions common to both maps, and some means of evaluating a region with respect to a resource. For example, it is important to know the distance between the site and the main utility lines.

The implementation of each of these functions must be considered when the encoding scheme is chosen. In later sections the relative merits of three encoding methods are considered together with the implications for the functions listed above. For now we will assume that these functions are available, and consider a possible command language.

4.4 A COMMAND LANGUAGE

Implementation of any system requires both determining a method of performing the functions, and making available to the user a means of carrying out these functions - a command language.

In the following discussion a command language is defined, and the functions available to the user by means of the language are noted.

4.4.1 SYNTAX OF THE COMMAND LANGUAGE

A relatively simple command language has been developed for use with the graphical display. It is intended to be used in conjunction with the GRID supervisor currently implemented on the display (Jacobsen et al, 1970). The following is a syntactical description of the commands as currently implemented.

The basic labels (e.g. DISPLAY) appearing in the syntax may be either words appearing on the screen or identifiers for various function keys. The simplest method (and in fact that adopted) is to have the words appear on the screen where they may be picked with the light pen. The alphanumeric symbols are available on the keyboard.

```
<COMMAND>::=<DISPLAY>|<LIST>|<DRAW>|<LINK>|<EVALUATE>|
            <ERASE>|<SCALE>|<PLOT>|<OVERLAY>|<SHADE>|
            <WINDOW>|<DELETE>|<CENTRE>|<RESTART>|<END>
<DISPLAY>::=DISPLAY<MAPNAME>
<LIST>::=LIST_MAPS {AVAILABLE | ON_SCREEN}
<DRAW>::=DRAW<MAPNAME>
<LINK>::=CATENATE<MAPNAME><VECTOR>
<EVALUATE>::=EVALUATE<X Y><MAPNAME>
<ERASE>::=ERASE<X Y>
<SCALE>::=SCALE {UP | DOWN}
<PLOT>::=PLOT
<OVERLAY>::=OVERLAY<MAPNAME>
<SHADE>::=SHADE<MAPNAME>
<WINDOW>::=WINDOW<INTEGER>
<CENTRE>::=CENTRE<X Y>
<DELETE>::=DELETE<MAPNAME>
<RESTART>::=RESTART
<END>::=STOP
<MAPNAME>::=<ABC><ABC><ABC><ABC><ABC><ABC><ABC><ABC>
<ABC>::=A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|
<INTEGER>::=0|1|2|3|4|5|6|7|8
<VECTOR>::=<VECTOR><LINE>|<LINE>
<X Y>::= a point indicated with the light pen
<LINE>::= that defined by Function Key 2 Status 0 in the
          GRID supervisor
```

4.4.2 SEMANTICS OF THE COMMAND LANGUAGE

For any drawing displayed on the screen there are two reference tags: 1) the centre of view and 2) the scale of presentation. We can consider the graphical display as a twelve inch square window showing one portion of a drawing. The point on the drawing which corresponds to the centre of the window is called the centre of view. The scale of presentation is the scale of the displayed drawing relative to the scale of the encoded drawing. Initially the centre of view is set to the centre of the encoded maps, and the scale of presentation is set to one.

With this in mind, let us look at each of the commands and see what facilities they offer the user.

COMMAND	RESULTING ACTION
<DISPLAY>	Display on the screen the map titled <MAPNAME>, where <MAPNAME> is picked, by means of the lightpen, from a list of mapnames displayed on the screen. The centre of view and the scale of presentation are not changed.
<LIST>	List on the screen the name(s) of the map(s) displayed if the word ON_SCREEN is picked, or, list on the screen the name(s) of the map(s) available (that is, in storage) if the word AVAILABLE is picked.

COMMAND	RESULTING ACTION
<DRAW>	Add to the list of mapnames the name <MAPNAME>, which is entered on the alphanumeric keyboard, and create a empty file for data which may be inserted by use of the <LINK> command.
<LINK>	Include with the data of a previously defined map <MAPNAME> the new data drawn on the screen. The user has the ability to add data to an existing map file.
<EVALUATE>	Evaluate the point <X Y> with respect to the resource <MAPNAME>. "Evaluation" is the application of some function which the user has previously supplied as a subroutine. The value of the function with respect to the point <X Y> is shown on the screen beside <MAPNAME>. For example, a function to determine the distance between a point and the nearest utility line may be given.
<ERASE>	Erase from the map file the region which contains the point <X Y>. <ERASE> may be used when there is only one map displayed on the screen. The outline of the region is deleted from the data file of that map.

COMMAND	RESULTING ACTION
<SCALE>	Change the scale of presentation. The centre of view remains unchanged. The scale of presentation is multiplied by a factor of two or one-half depending on whether UP or DOWN is picked with the lightpen.
<PLOT>	Plot on the Calcomp Plotter a hard copy of the items currently displayed on the screen.
<OVERLAY>	Add to the items currently displayed on the screen the map <MAPNAME>. The regions common to both are determined and filed as a separate map OVERLAY which may also be displayed.
<SHADE>	Shade the regions of the map <MAPNAME> which are currently displayed on the screen. The regions are filled in with straight line segments.
<WINDOW>	Set the centre of view to the centre of the section chosen and multiply the scale of presentation by a factor of three. The portion of the screen which is set aside for displaying maps is divided into nine sections (see figure 7.1). The integer

COMMAND	RESULTING ACTION
	designating the section the user wishes to see must be typed on the keyboard.
<CENTRE>	Set the centre of view to the point <X Y>. The scale of presentation remains unchanged.
<DELETE>	Delete from the system all information concerning the map <MAPNAME>.
<RESTART>	Set the scale of presentation to one. Set the centre of view to the centre of the encoded maps. Clear the screen.
<END>	Terminate the session.

4.5 AN EXAMPLE

The facilities made available through use of the command language may best be understood from an example of how it might actually be used by the planner. The example is divided into three sessions. In terms of Dror's six planning tasks, session one would be to complete tasks two and three, session two for task four, and session three for task five. The commands which may be most used in a part of a session are given in brackets at the end of that part. At the end of session one the planner has a map showing the locations of the various alternative land sites. At this point he may consult the university authorities as to whether any of the areas are reserved for other future projects. In

the second session he determines a likely site based on the remaining alternatives. After additional study of information on this site, he is better able to give a true evaluation of this site in relation to the proposed project. He may decide to consider alternatives other than the current one. One additional method of study is to add or delete from the resource maps those resources created or consumed by the building under consideration, for example, pedestrian pathways may be increased, and the amount of open space decreased.

4.5.1 SESSION ONE

(1) Check out each of the resource maps and the base map. Make any corrections deemed necessary. (<ERASE>,<DISPLAY>,<LINK>)

(2) Insert any extra maps necessary for the study. (<DRAW>,<LINK>)

(3) Determine which regions are possible building sites and draw one map containing these regions. This may be done by viewing the maps as a series of overlays and considering how well various regions fulfil the requirements of the new structure. (<OVERLAY>,<WINDOW>,<SCALE>,<DRAW>, <LINK>)

(4) Make a hard copy of this map for discussion with university officials. (<PLOT>)

4.5.2 SESSION TWO

(1) Erase from the map of alternatives those areas ruled out by the administration. (<ERASE>)

(2) Evaluate the remaining alternatives to determine how well they fulfil the planning objectives. (<EVALUATE>)

(3) Eliminate those alternatives which are deemed undesirable based upon the above evaluation. (<ERASE>)

(4) Obtain a hard copy of the optimal alternative. (<PLOT>)

4.5.3 SESSION THREE

(1) Incorporate into each resource map the changes that would result if the building were to be constructed at the site under consideration. (<LINK>,<ERASE>)

(2) Overlay the resources to check for major faults which might result. (<OVERLAY>,<WINDOW>,<SCALE>)

(3) Choose the current site as the building site or return to session two and choose another for study.

4.6 SUMMARY

The system deals almost entirely with maps. The original maps must be encoded and the encoded data stored on disc or magnetic tape. The planner, when he is using the system, is in effect constantly retrieving this data and converting it to the form required for display on a Cathode Ray Tube. Before the data is displayed, there may be functions to be applied to the encoded data. In this system

these functions are: scaling, overlaying two maps, and windowing, that is viewing only a portion of the total map(s). The user must also have the facility to draw maps on the display and have the proper files created and maintained.

4.7 CONCLUSION

The author has described one approach towards implementing the Sieve Process in a graphical display environment. A command language has been defined to use at the display terminal. Several important functions manipulating map data have been specified as major requirements of the system. The method of implementing these functions is heavily dependent on the manner in which the map data is represented in the system. The next step then is to consider the possible methods of representation, how any particular method affects the implementation of the desired functions, and hardware limitations. As a result of these considerations one encoding method must be chosen.

CHAPTER V

MAP ENCODING

5.1 INTRODUCTION

The basis of the inventory for the Sieve Process is the map form. Three methods of representing map data in digital computers have been investigated: storage of XY coordinates of points (as illustrated by Mezei, 1968b), Chain-encoding (Freeman, 1961a, 1961b) and Skeleton encoding (Rosenfeld and Pfaltz, 1966; Pfaltz and Rosenfeld, 1967; Pfaltz, Snively and Rosenfeld, 1968). Mezei's SPARTA package allows artists unfamiliar with computer languages to use the computer as an artistic tool (Mezei, 1967, 1968a). Freeman says chain-encoded data is best when calculating such things as the area of arbitrary regions. Rosenfeld and Pfaltz have used the skeleton method to obtain data representations of maps for urban studies. It has been found in this investigation that no one method is superior overall and the author concludes that the choice of the most suitable method depends very much on what one wants to do.

For the purposes of this discussion the following definitions are used: An arbitrary line is a line not necessarily subject to mathematical formulation. A region is a connected planar area bounded by one or more closed lines. A map is a set of arbitrary lines and regions.

In all three methods arbitrary lines are approximated by a series of straight line increments.

5.2 CRITERIA

When we consider any application involving line drawings, four factors must be considered when choosing an encoding method. These are: 1) storage requirements for the data, 2) the degree of exactness (fineness of resolution) demanded in the reconstructed (or displayed) drawing, 3) time required to convert the encoded representation to a CRT or plot display, 4) the type of processing required to obtain the desired answers.

Storage requirements for encoded data are obviously important. The fineness of resolution demanded by the programmer affects the storage requirements; the increase in storage needed is heavily dependent on the method used. The time used to set-up data for a CRT display (if this is necessary) could be critical since the set-up program may be frequently used. Examples of processing of line drawings are: determining 1) area, 2) perimeter, or 3) the intersection of regions. It has been found by the author that the time required for these operations varies from method to method. One method (of representing data) may be markedly superior if we consider the speed with which some operation may be applied. However, if we consider another operation, the position may be reversed. For each of several operations likely to be applied to map data the methods are compared in chart form in a later section. For quantitative comparisons, the equipment presently available at the University of Alberta (an IBM 360/67 system, Control Data

GRID display and a model 663 Calcomp plotter) is used as a basis. The GRID CRT screen is considered as a rectangular grid of 1024 x 1024 points (approx. 12 inches x 12 inches) and vectors can be drawn between any pair of points. The Calcomp plotter allows vectors to be drawn in any of sixteen directions to a resolution of .0025 inches.

5.3 XY COORDINATE METHOD

Each of the line segments representing an arbitrary line may be encoded by storing the x and y coordinates of its endpoints. Generally this would mean that one 32-bit word of storage in the IBM 360/67 would be required for each of these coordinates, or two words per point. A curved line is encoded as a series of straight line segments whose lengths are chosen at the discretion of the user at the time of encoding. Each curve, then, may be stored in a 2xN array where N is the number of points used to describe the curve. Two possibilities exist for economizing on storage use:

(1) a pair of x and y coordinates may be packed into one word provided

$$| x*10^i | \leq 2^{15}-1 \text{ and}$$

$$| y*10^i | \leq 2^{15}-1$$

for all x,y, and where i is the number of decimal places required for each coordinate, for example, assuming i=2

$$\text{then } |x| \leq 327.67$$

$$\text{and } |y| \leq 327.67$$

(2) interpolation may be used to reconstruct the outline. Fewer points would then be required per line. However, the accuracy of representation is decreased.

An example of an application for which xy encoded data is very suitable is one in which scaling of drawings is frequently required. The representation also, for example, allows rotation through a variable number of degrees in a straightforward manner. However, the representation is unsuitable if, for example, the shading of arbitrary regions with straight line segments is required. Shading requires the repeated calculation of the points of intersection of a line with the regions given, a process which requires much compute time for data encoded in this form.

5.4 CHAIN ENCODING

The idea of Chain-encoding is as follows: Any point x may be considered as the center of a rectangular grid (as shown in figure 5.1). From this point one can move to any of eight neighbouring points, labelled 0 to 7. Thus any line may be approximated by a series of vectors of two standard lengths (1 unit and $\sqrt{2}$ units) and eight standard directions, and may therefore be represented by a series of integers ranging from 0 to 7, e.g. figure 5.2. These integer increments may be packed ten to a word in the 360/67.

When employing this method, the programmer must decide on the length of the basic unit in the chain (i.e. the grid

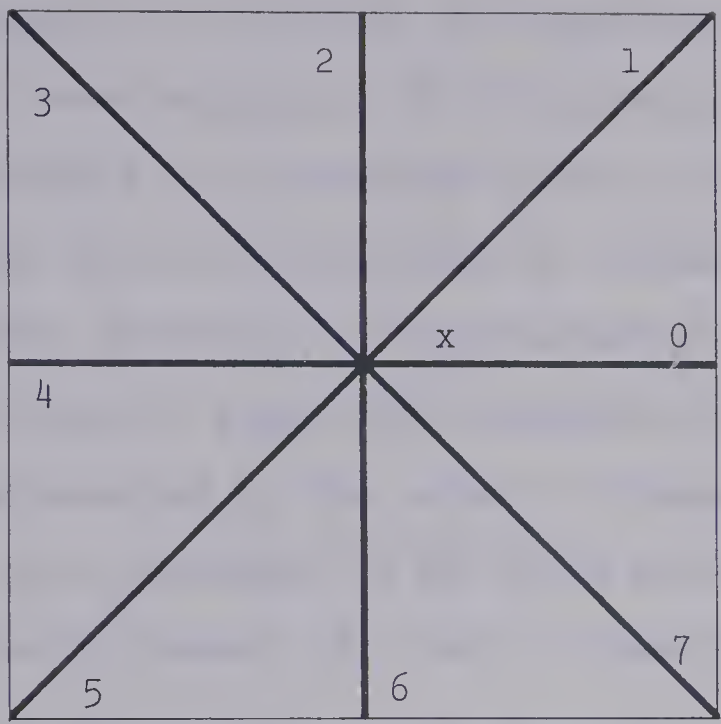
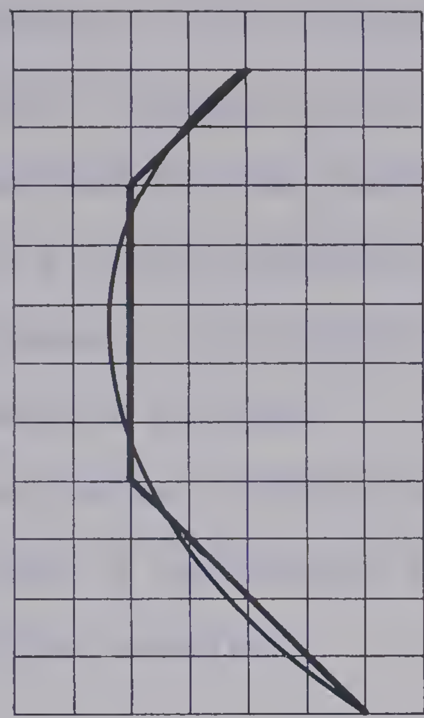


Figure 5.1

Chain-encoding grid



The curve is encoded as 3333222211

Figure 5.2

Sample Chain-encoding data

size). The length of the basic unit may be decided on by considering the degree of exactness required in the reconstructed drawing, and the length of the smallest segment to be encoded. Storage requirements may be reduced by using "generating functions" for straight lines or standard curves, as described by Freeman (1961a).

Chain encoding is most suitable, for example, when representing fine outlines of arbitrary figures, since the line segments representing the curve must necessarily be small.

This mode of representation allows, for example, determination of the area of a region in a straightforward manner. However, this representation is unsuitable, for

example, if rotation is required. Rotation is only possible for even multiples of 45 degrees; rotation through any odd multiple of 45 degrees results in distortion of the figure. This can be illustrated as follows: If a vector component, whose direction is represented by the value 1, is rotated through an angle of 45 degrees its direction is then represented by the value 2. However the value 1 represents a line increment of $\sqrt{2}$ units and the value 2 represents a line increment of 1 unit. Thus distortion results.

5.5 SKELETON ENCODING METHOD

A region may be encoded as a set of "skeleton" points and associated radii. The following definitions are necessary:

1. The "distance" between two points P_{ij} and P_{km} is defined as

$$d(P_{ij}\ ,\ P_{km}) = |i-k|+|j-m|$$

2. A "neighbourhood" of radius r of a point P_{ij} is the set of all points P such that

$$d(P_{ij}\ ,P\ ) \leq r$$

3. A neighbourhood is a "maximal neighbourhood" if no other neighbourhoods are properly contained within it. (Pfaltz and Rosenfeld 1967).

Using an optical scanner, a picture (transparency or equivalent) may be digitized as an array A with elements a_{ij}, where (i,j) represent Cartesian coordinates of a point and a_{ij} is the density, or "grey level", of the digitized image

over a small region surrounding that point. Note that if A is of dimension (1024,1024) a one-to-one mapping of A onto the GRID screen is possible. Given this digitized image the maximal neighbourhoods may be determined. (An algorithm for this is given by Rosenfeld and Pfaltz.) Thus there exists a set of coordinates and radii which describe the input. The locus of these coordinates is called the "skeleton".

If each of these three values x,y and r is ≤1024 it is possible to pack all three into a single 32 bit word in the IBM 360/67.

Algorithms are available (Rosenfeld and Pfaltz, 1966; Pfaltz and Rosenfeld, 1967; Pfaltz, Snively and Rosenfeld, 1968) which describe in some detail the processing required to obtain the skeleton, or to obtain the outline of the region from the skeleton.

This method is the best of the three discussed if, for example, one wishes to determine the intersection of arbitrary regions. However, obtaining the area of any given region requires more processing time than either of the other methods.

5.6 COMPARISON IN CHART FORM

Several general comparisons are possible between the three methods previously described. These are given in chart form below.

Following the discussion of each function the author has indicated how the methods compare relative to each other.

STORAGE REQUIREMENTS			
DATA REPRESENTED BY / FUNCTION	XY COORDINATES	CHAIN ENCODING	SKELETON ENCODING
Storage requirements when curvature is small	Two words of memory in the 360/67 per point. May be most economical if packing of data into one word is possible or interpolation is acceptable.	Ten vectors may be packed into one word of memory in the 360/67 not economical if a small basic unit of length is required.	Each neighbourhood has three values associated with it. Not economical in this situation if a small basic unit is required.
	good	relatively poor	relatively poor
Storage requirements when curvature is large	Storage used is generally excessive in this case, due to the number of points required to outline the curves.	Storage is used economically in this case because a small basic unit of length is demanded.	Storage is roughly equivalent to that required for chain encoding.
	relatively poor	good	good
Storage requirements for encoding open curves	Open curves are encoded as a set of points.	Open curves are encoded as a set of vectors.	Open curves are encoded as a set of coordinates with radius zero - i.e. every grid point on the curve is required.
	good	good	relatively poor

OUTPUT			
DATA REPRESENTED BY / OUTPUT FOR	XY COORDINATES	CHAIN ENCODING	SKELETON ENCODING
Incremental plotter	Data is easily processed.	Data is easily processed.	Considerable processing time is required to obtain an XY coordinate outline.
	good	good	relatively poor
CRT display file	Data is easily converted to the "absolute" mode for GRID.	Data is easily converted to the "relative" mode for GRID.	Considerable processing time is required to obtain an XY coordinate outline.
	good	good	relatively poor

FUNCTIONS OF DATA MANIPULATION

DATA REPRESENTED BY / FUNCTION	XY COORDINATES	CHAIN ENCODING	SKELETON ENCODING
Determining the area of a region	A straight forward algorithm is available.	A straight forward algorithm is available.	Must first convert to outline form then use same method as for XY coordinate encoding, <u>or</u> sum areas of neighbourhoods and subtract the overlap. Either way, relatively much processing is required.
	good	good	relatively poor
Determining the perimeter of a region	Sum the distance between consecutive points.	A straight forward algorithm is available.	The outline must be obtained first, then the distance between consecutive points can be determined and summed.
	good	good	relatively poor

FUNCTIONS OF DATA MANIPULATION

DATA REPRESENTED BY / FUNCTION	XY COORDINATES	CHAIN ENCODING	SKELETON ENCODING
Determining the intersection of two regions	The points of intersection are easily determined. To obtain the outline of the common region requires some processing.	The points of intersection are easily determined. To obtain the outline of the common region requires some processing.	Minimal processing gives a skeleton representation of the common region of intersection - however the neighbourhoods need not all be maximal, so some storage may be wasted.
	relatively poor	relatively poor	good
Rotating figures	Data may be rotated through a variable number of degrees without distortion.	Data may be rotated only through multiples of 45 degrees; and only multiples of 90 degrees without distortion.	May be rotated through a variable number of degrees but only through multiples of 90 degrees without distortion.
	good	relatively poor	relatively poor

FUNCTIONS OF DATA MANIPULATION

DATA REPRESENTED BY / FUNCTION	XY COORDINATES	CHAIN ENCODING	SKELETON ENCODING
Scaling figures (up or down)	Data may be scaled in a straight-forward manner.	Scaling involves changing the basic unit of length, or simply performing a many for one exchange of vectors - either way implies some limitation on the range of scaling possible.	Scaling involves a change of the basic unit or radius, and scaling the skeleton outline appropriately.
	good	relatively poor	good
Shading of regions	Shading involves repeated testing for the intersection of lines with regions.	Shading involves repeated testing for the intersection of lines with regions.	Shading may be accomplished with minimum processing time using this method.
	relatively poor	relatively poor	good

5.7 CONCLUSION

Three possible digital representations of arbitrary line drawings have been considered. Several functions, those which are most likely to be used in a general graphics program using line drawings, have been considered in relation to each representation. From the comparison chart it is evident that no one method is best for all functions. For any application, the choice of encoding method depends heavily upon which of these functions the user requires.

With this in mind, one may proceed to discuss in greater detail the suitability of these methods for the functions required to implement the Sieve Process.

CHAPTER VI

NOTES ON ALGORITHMS FOR FINDING THE INTERSECTION OF REGIONS

6.1 INTRODUCTION

When considering the function of windowing for display we see that it is equivalent to the following situation:

Given any two regions A and B, for our purposes there are four possible relations between them. These are illustrated in figure 6.1.

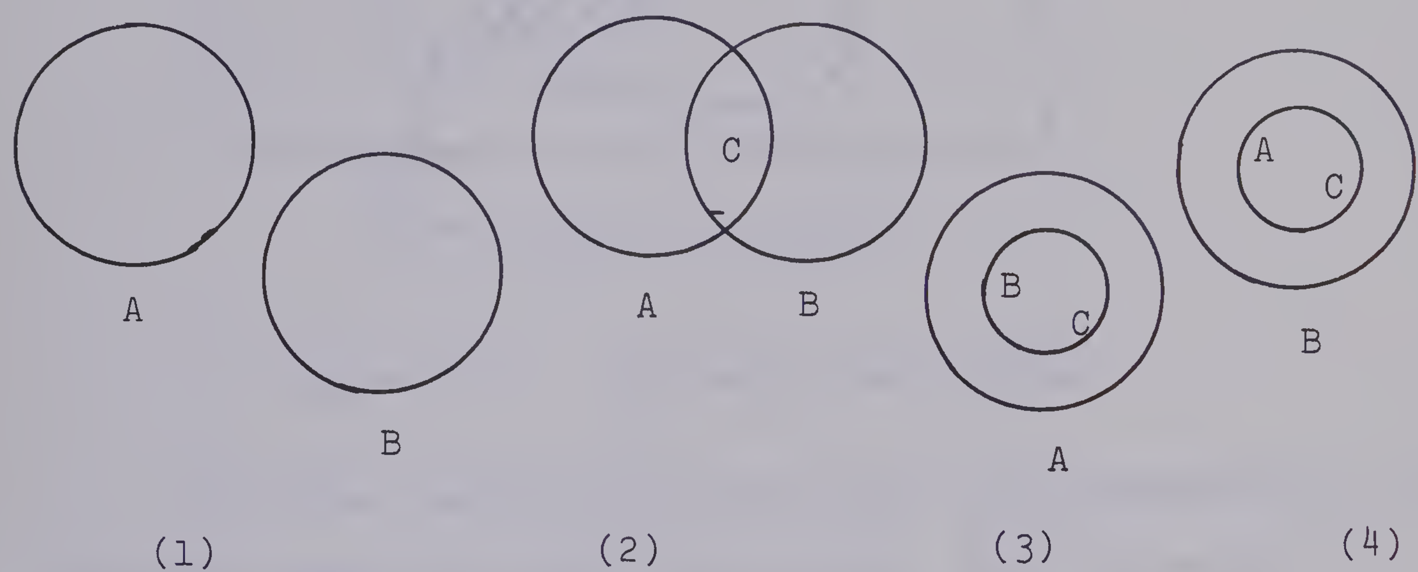


Figure 6.1 $A \cap B$

Algebraically they are:

(1) $A \cap B = \emptyset$
(2) $A \cap B = C$
(3) $A \cap B = B = C$
(4) $A \cap B = A = C$

For windowing for display, there is one map A and a window of view B. The result of the intersection $A \cap B$ is C, that portion of the map which is to be displayed on the screen (figure 6.2).

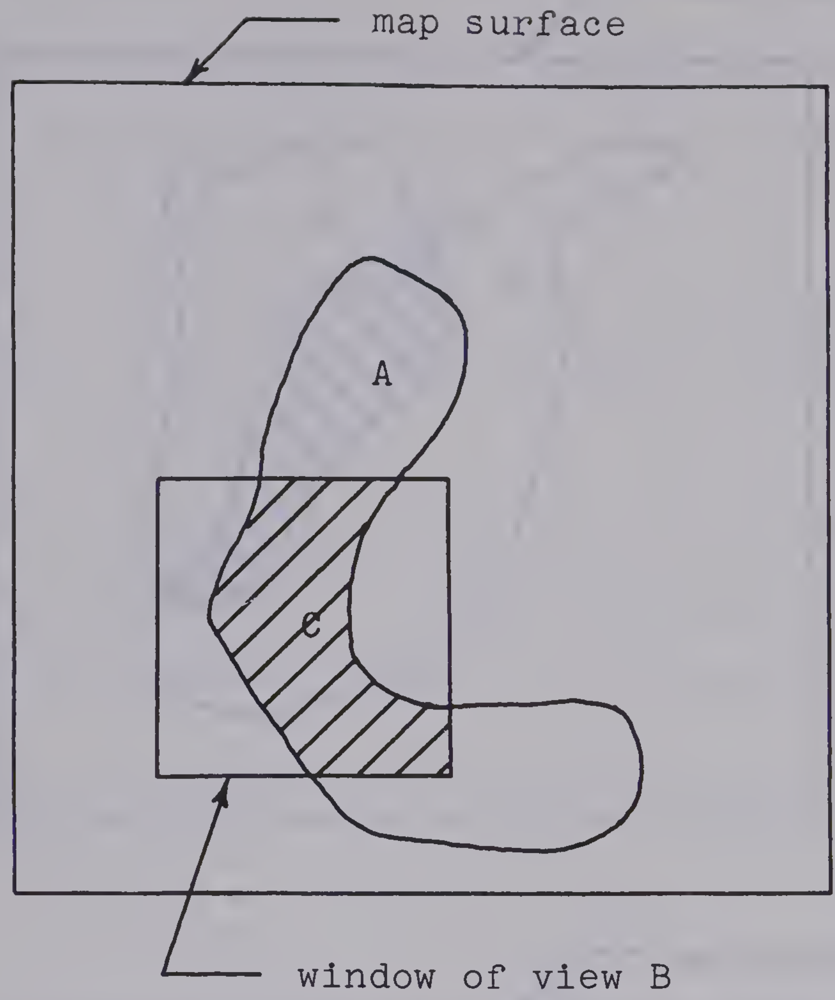


Figure 6.2 Windowing for display

For the function of overlaying, the intersection E of two maps A and D is first calculated. Then the region C must be determined where $E \cap B = C$, B being the window of view. It is the region C that is displayed on the screen (figure 6.3).

Thus, the implementation of two important functions for the Sieve Process requires some method for calculating the intersection of regions. Due to the extensive use of this function, we shall consider it in relation to the three types of data representation in some detail.

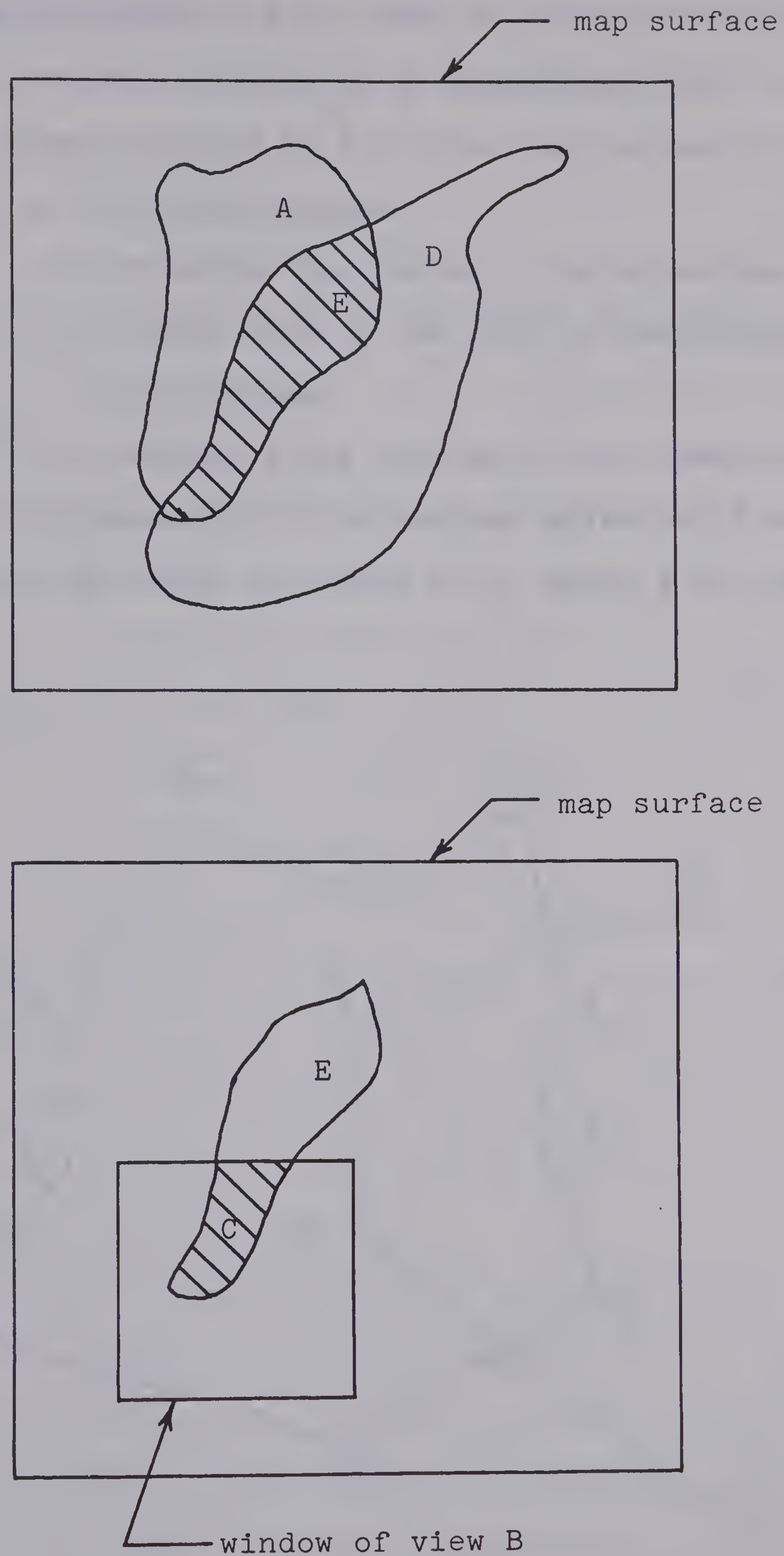


Figure 6.3   Overlaying two maps

6.2 AN ALGORITHM FOR USE WITH XY COORDINATE DATA

For data encoded in XY coordinate form, there are three steps required to determine the outline of regions common to two given regions:

(1) Determine the points of intersection.

(2) Insert them in the list of coordinates describing the regions.

(3) Determine the outline of the common regions.

For example, if the regions given are A and B (which intersect as shown in figure 6.4), where A is identified by

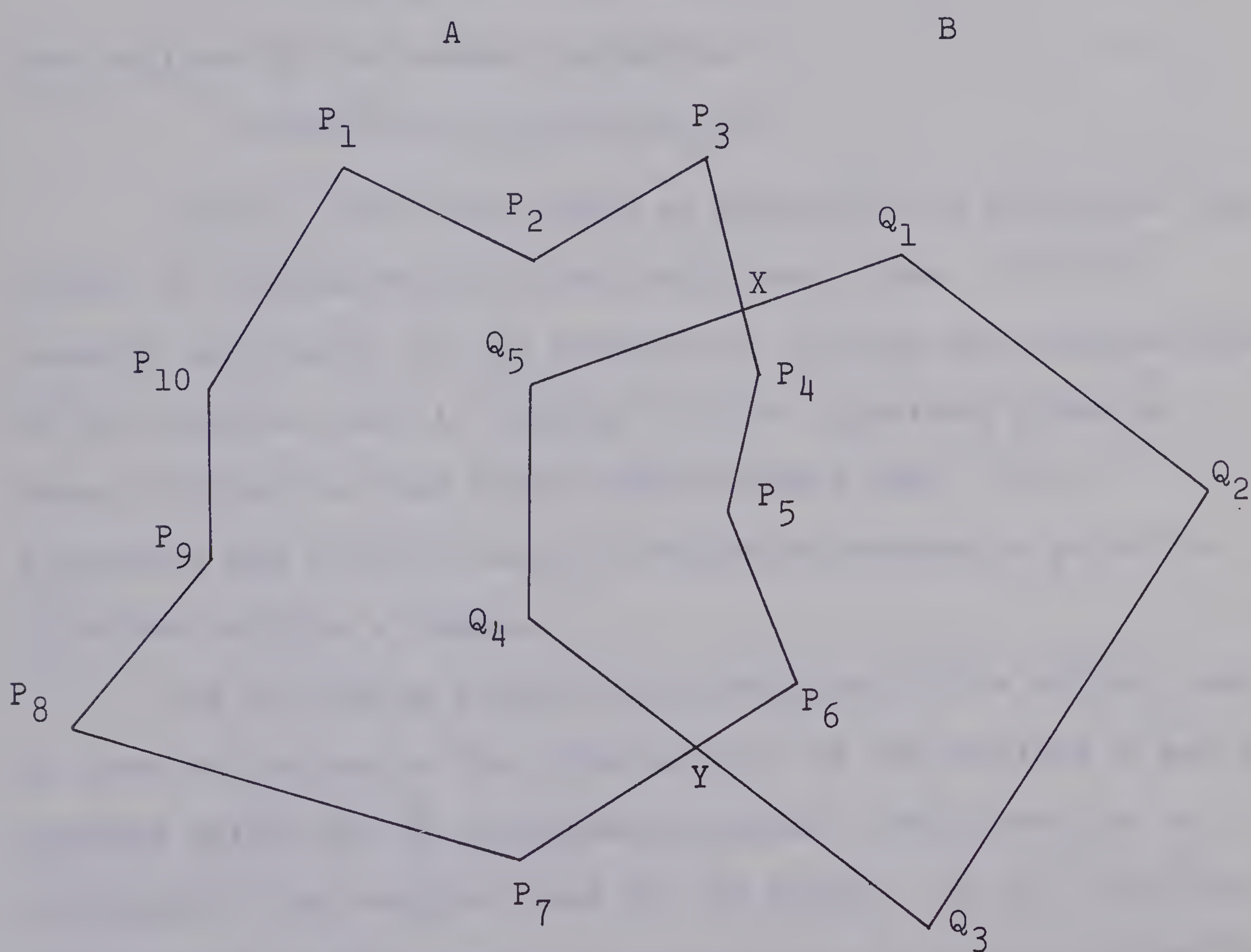


Figure 6.4 A∩B

the set of points

$$P=(P_1,P_2,P_3,P_4,\ldots P_9,P_{10},P_1),$$

and B is identified by the set of points

$$Q=(Q_1,Q_2,Q_3,Q_4,Q_5,Q_1),$$

then, after points X and Y have been determined, they must be inserted in the sets P and Q. Thus the set describing A becomes

$$P=(P_1,P_2,P_3,X,P_4,P_5,P_6,Y,P_7,P_8,P_9,P_{10},P_1)$$

and the set describing B is

$$Q=(Q_1,Q_2,Q_3,Y,Q_4,Q_5,X,Q_1).$$

The outline of the common region is

$$R=(X,P_4,P_5,P_6,Y,Q_4,Q_5,X).$$

Loomis (1965) has given an algorithm to determine the points of intersection of two arbitrary lines. This is equally applicable to the problem of finding the intersection of two regions, and is similar to the algorithm given by Morse (1968b) for use with chain-encoded data. The algorithm may also be used to determine whether a point is contained within a region.

The following algorithm, developed by the author, may be used to determine the intersection of two regions A and B encoded using the XY coordinate method. The direction of encoding of the regions need not be known. If the direction of encoding of both regions is clockwise, a simpler algorithm is available and will be discussed later in the section.

STEP 1 Let N be the number of points of intersection of regions A and B. If $N=0$ go to STEP 12. Assume $N>0$.

STEP 2 Let $X=(X_1,X_2,...X_N)$ be the points of intersection with respect to region A, and let $X'=(X'_1,X'_2,...X'_N)$ be the points of intersection with respect to region B, where $i=1,N$ indicates that the point X_i or X'_i is the i^{th} point in the set of coordinates describing region A or B respectively.

STEP 3 For each X_i determine if either point X_i-1* or X_i+1 is contained inside or on the boundary of region B. If neither point satisfies this condition then the regions are tangential to each other at the point of intersection; point X_i is deleted from the set X, and X'_j, which corresponds to X_i, is deleted from the set X'.

STEP 4 Thus, there remains the set $X=(X_1,X_2,...X_K)$ and $X'=(X'_1,X'_2,...X'_K)$ where $K\geq 0$. If $K=0$ go to STEP 12. Assume $K>0$.

STEP 5 Set $i=1$ and set $s=1$.

STEP 6 Determine if the point X_i+1 is outside region B. If so, set $j=i-1$, if not, set $j=i+1$. If $j\geq K$ set $j=1$, if $j=0$ set $j=K$.

---

* X_i-1 refers to the point before X_i in the description of A, after X_i is inserted into the set of points describing A. For example, from figure 6.4 if $Y = X_2$, then $X_2+1 = P_7$ and $X_2-1 = P_6$.

STEP 7 The outline of a region consists in part of the points from X_i to X_j. Delete X_i and X_j from the set X.

STEP 8 X_j corresponds to X'_b of set X'. Determine if the point X'_b+1 is outside region A. If so set d=b-1 if not set d=b+1. If d>K set d=1, if d=0 set d=K.

STEP 9 The outline of a region consists in part of points from X'_b to X'_d.

STEP 10 X'_d corresponds to X_m of set X. If m=s one region is outlined, go to STEP 11. If m≠s the outline is not complete, set i=m and go to STEP 6.

STEP 11 If there are no elements remaining in set X the process is complete (EXIT). If there are elements left, set i to the index of the first such element in X and set s=i. Go to STEP 6.

STEP 12 Three possibilities remain to be examined:

(1) A is contained in B,

(2) B is contained in A, or

(3) A and B have no common region.

If any point of A is inside B, then the region of intersection is A; else if any point of B is inside A, then the region of intersection is B; else there is no region of intersection.

6.3 AN ALGORITHM FOR USE WITH CHAIN-ENCODED DATA

Where data is chain-encoded two steps are required to obtain the intersection of two regions:

(1) Obtain the points of intersection, if any.

(2) Determine the outline of the common region or regions, if any.

The points of intersection may be determined in any one or more of three ways; these are discussed by Morse (1968b) and Freeman (1961b).

It can be shown that if both regions are closed, and encoded in a clockwise direction, then the following algorithm determines the outline of all regions common to both the given regions.

STEP 1 Let N=number of points of intersection. If N=0 go to STEP 11. Assume N>0.

STEP 2 Let $P=(P_1,P_2,...P_N)$ be the points of intersection with respect to region A, and let $P'=(P'_1,P'_2,...P'_N)$ be the points of intersection with respect to region B, where i(i=1,N) indicates that the point of intersection P_i or P'_i is at the end of the i^{th} vector in the chain describing region A or B respectively.

STEP 3 For each P_i, (i=1,N), determine whether either point P_i-1 or P_i+1 is contained inside, or on the boundary of, region B. If neither point satisfies this condition, then the regions are tangential to each other at the point of intersection. The

point P_i is deleted from the set P and point P'_j, which corresponds to P_i, is deleted from the set P'.

STEP 4 Thus there remain points $(P_1, P_2, \ldots P_K)$ and $(P'_1, P'_2, \ldots P'_K)$ where $K \geq 0$. If $K=0$ go to STEP 11.

STEP 5 Determine if the point P_1+1 is contained within region B. If the point $P_1+1=P_2$, then consider the midpoint between P_1 and P_2. If the point is contained within, or on the boundary of, region B then set $i=1$. If the point is not contained in region B, set $i=2$.

STEP 6 Set P_i as the initial point in the chain. Set $m=i$.

STEP 7 Set $d=i+1$. If $d>K$ set $d=1$. Copy the chain from P_i to P_d.

STEP 8 The point P_d corresponds to P'_j, say, of region B. Set $f=j+1$. If $f>K$ set $f=1$. Copy the chain from P'_j to P'_f. Delete P_i and P_d from the list of points of intersection.

STEP 9 P'_f corresponds to P_g, say, of region A. If $g=m$ one region has been outlined. If $g \neq m$ set $i=g$ and go to STEP 7.

STEP 10 If there are no points of intersection remaining in P, the complete region of intersection has been delineated (EXIT). If there are elements remaining, set i to the index of the first such element P_j where $j>m$.

STEP 11 Three possibilities remain to be examined:

(1) A is contained in B,

(2) B is contained in A, or

(3) A and B have no common region.

If any point of A is inside B, then the region of intersection is A; else if any point of B is inside A, then the region of intersection is B; else there is no region of intersection.

Note that, using the above algorithm, common boundaries are given as regions of intersection; for example, from figure 6.5, the boundary XY is considered a common region.

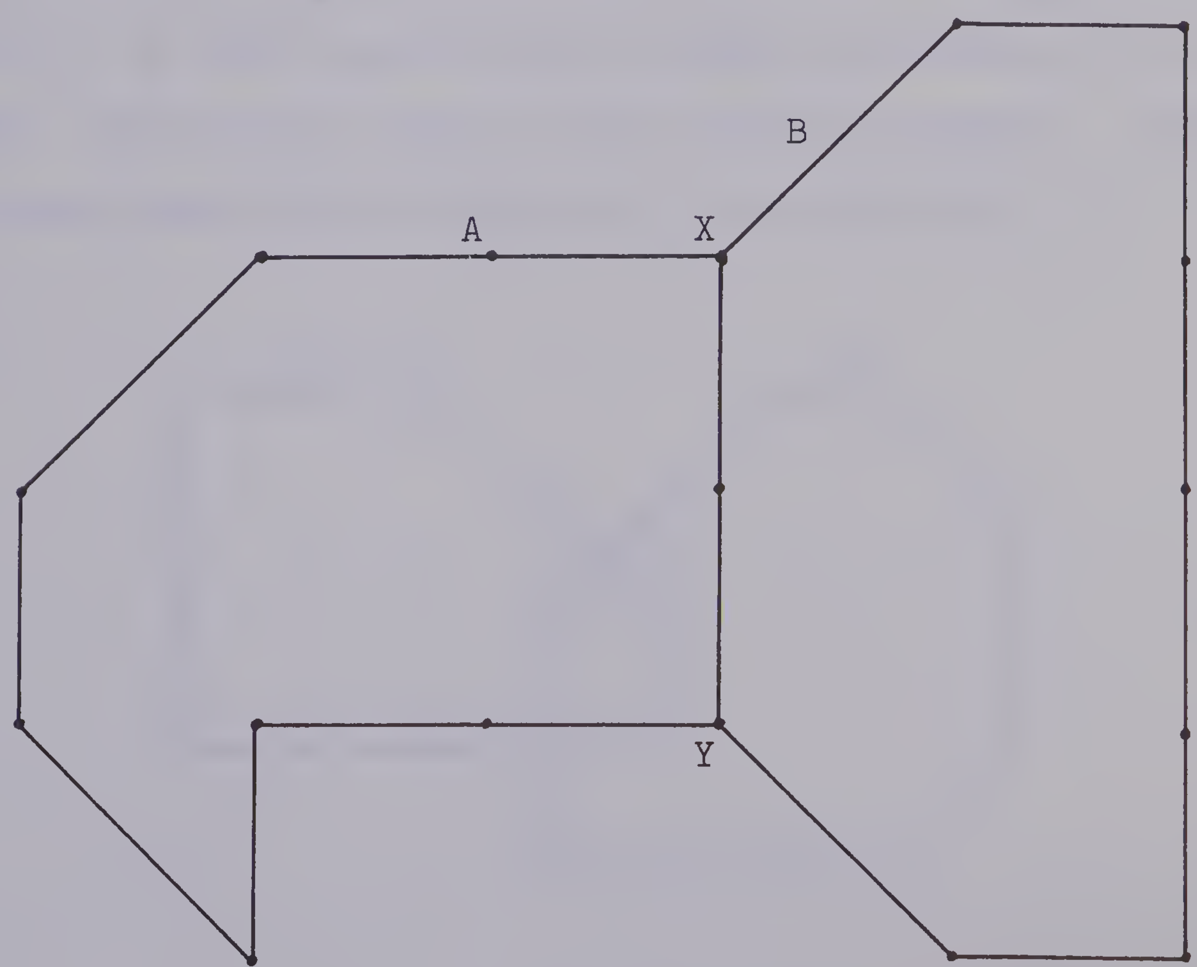


Figure 6.5 A∩B with chain-encoded data

If the direction of encoding of the regions is not known, or is not consistent, then more processing is required. In fact the algorithm outlined for the XY coordinate data may be used if the words "the points" in steps 7 and 9 are changed to "chain of vectors".

Similarly, for XY coordinate encoded data, if the regions are encoded in clockwise order the algorithm given above may be used to determine the region of intersection, after the points of intersection have been inserted in the outline of regions A and B. However, "the chain" in this case is a list of coordinate pairs.

There is a problem with chain-encoded data in that a point of intersection may not occur at the endpoint of a vector. In this case one must change one element of the chain. This can be seen in the following example. Given regions A and B as in Figure 6.6, the algorithm for

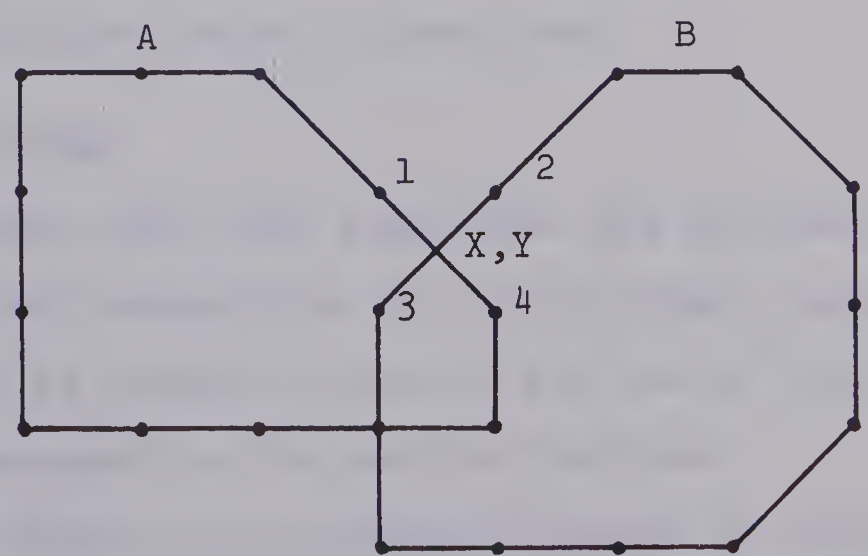


Figure 6.6 A∩B problem with chain-encoded data

obtaining the points of intersection yields the true X and Y coordinates. These are not endpoints of any member of the chains for A or B, and cannot be made endpoints of any member of the chain C. Rather, either chain A or chain B must be altered so that the intersection occurs at one of the four vertices V=(1,2,3,4). Only then can one proceed to determine the outline of the common region.

6.4 AN ALGORITHM FOR USE WITH SKELETON ENCODED DATA

The region $A \cap B$ is easily determined if the regions A and B have been encoded by the Skeleton method. Pfaltz and Rosenfeld (1967) have outlined an algorithm for determining the skeleton data of the common region; however, they point out that the result does not necessarily give maximal neighbourhoods, and consequently excessive storage may be used. Maximal neighbourhoods may be obtained by converting the result to a point-matrix picture and re-encoding the matrix using the standard algorithms.

6.5 CONCLUSION

We have seen that algorithms are available for obtaining the intersection of two arbitrary regions. The regions may be encoded by use of any one of the three methods discussed in the previous section.

The amount of processing required to calculate the outline of the region of intersection is substantially more for XY coordinate or chain-encoded data than for skeleton data. However, it must be noted that the result using

skeleton encoded data may be inexact (that is, maximal neighbourhoods do not necessarily result) unless it is decoded and re-encoded to obtain maximal neighbourhoods.

We have seen how the intersection of two regions may be determined, given data in any one of the three representations discussed previously. This function is required in order to implement several functions for the Sieve Process. We are now able to consider how these functions may be best implemented, and in so doing decide on an encoding method for the process.

CHAPTER VII

PROBLEMS OF IMPLEMENTATION

7.1 INTRODUCTION

Three problems will be discussed in this section:

(1) which data representation should be used to obtain minimum processing time;

(2) which data representation should be used to obtain minimum storage requirements;

(3) limitations of the GRID.

7.2 THE THREE ENCODING METHODS IN RELATION TO THE SIEVE PROCESS - MINIMIZING PROCESSING TIME

Since the amount of processing required to execute most of the actions outlined in the command language depends on the type of data representation used, it is important to consider the amount of processing required for commands which may be used repeatedly. Such commands are: <DISPLAY>,<OVERLAY>,<SCALE>,<WINDOW>,<LINK>, and <ERASE>.

A package (GRIDSUB) for use with the GRID has been developed by W. H. Huen (1969). GRIDSUB maintains in a "relative mode" all items to be displayed, that is, all coordinates of a line, except the initial coordinates, are determined relative to the previous coordinate rather than to an absolute origin. This enables the same item to be displayed at various positions on the screen with a minimum of processing and storage requirements.

7.2.1 <DISPLAY>

Two steps are required to display a map on the CRT:

(1) obtain the outline of the regions of the map which will be displayed on the screen (that is, the intersection of the map with the window of view),

(2) set up the files required for transmission to GRID.

The previous chapter discussed the intersection process. GRIDSUB files are set up from an array of coordinate pairs, the coordinates may be determined either with respect to an absolute origin, or with respect to the previous coordinate pair. To obtain this array requires no processing of data encoded as XY coordinates, minor processing for chain-encoded data, and much more processing for data encoded by the skeleton method. The result is that to accomplish step 1 skeleton encoded data is the best choice; to accomplish step 2 skeleton encoded data is the poorest choice. From the point of view of minimizing processing no one data representation is best, any one of the three is acceptable.

7.2.2 <OVERLAY>

The overlay facility is equivalent to three consecutive steps:

(1) obtain the outline of the regions common to the image on the screen and the map <MAPNAME> picked with the lightpen,

(2) store this data as a map which may be displayed later,

(3) add to the image on the screen the map used in step 1 above.

Steps 1 and 3 above both require a facility for determining the intersection of regions. Step 2, the storage of encoded data, is insignificant compared to steps 1 and 3. From the point of view of minimum processing time the skeleton data appears the best to use to implement this command, as it may be processed fastest in steps 1 and 3.

7.2.3 <SCALE>

XY coordinate encoded data may be scaled up or down in a straightforward manner. Skeleton encoded data may be scaled with little processing in a simple manner, namely, scaling the skeleton and the unit of radius appropriately. Somewhat more processing is required, however, if the unit of radius is to remain unchanged.

With chain-encoded data, scaling about the initial point may be done by simply changing the length of the basic unit. Scaling about any reference point involves changing the length of the basic unit, and a linear translation of the initial point. For minimum processing time, the chain-encoded data is the best choice; the XY coordinate data is the poorest choice.

7.2.4 <WINDOW>

The display screen, or at least that part of it dedicated to displaying maps, is divided into nine sections. These are numbered from zero through eight as shown in figure 7.1.

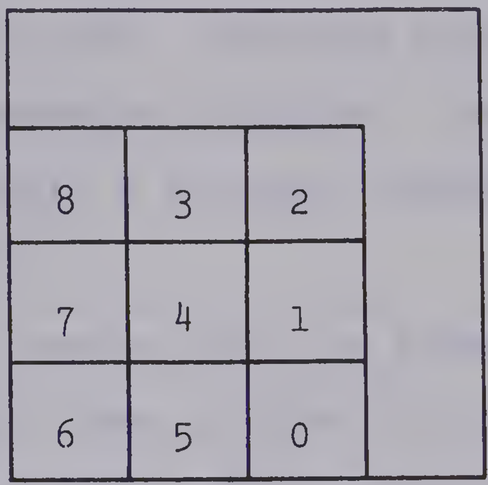


Figure 7.1 Map sections on the CRT

At any time in the session the planner is able to choose, and to have displayed at full screen size, any one of the nine windows of the image being displayed. Note that this is equivalent to changing both the scale of presentation and the centre of view, followed by displaying each of the maps currently shown on the screen. In this case any of the three data representations is acceptable, as approximately the same amount of processing time is required regardless of the representation chosen.

7.2.5 <LINK>

A drawing supervisor currently being considered for the GRID terminal would save absolute XY coordinates. Hence, if one "draws" a map or a portion of a map on the display, the XY coordinate data can be simply saved in the 360/67 with no more processing required to convert the data, if XY coordinate data is used. Moderate processing is required to obtain a chain-encoded outline. Substantial processing is required to obtain a skeleton representation of the map.

7.2.6 <ERASE>

To erase a region from the screen the user picks the command word ERASE, then a point <X Y>. The region, of the map currently displayed, which contains the point is then determined and deleted from the map data. The processing required to determine whether a point <X Y> is inside a given region is equivalent to that required to determine the points of intersection of two regions, where one region is a straight line. Therefore least processing would be required if data were skeleton encoded. The processing required for actual deletion of a region from a map is roughly equivalent for each of the three methods.

7.2.7 SUMMARY

Several of the more important functions that are required to implement the Sieve Process have been investigated. In terms of least processing time, the best choice for encoding data is, in most cases, the skeleton method. This is because the intersection of regions must be

found in many of the functions. There is little to choose between the other two representations, as both generally require more processing than the skeleton data.

7.3 MINIMIZING STORAGE REQUIREMENTS

Storage requirements for map data are dependent on the degree of exactness demanded in the reconstructed drawing. Given arbitrary curves, for a high degree of exactness XY coordinate data requires excessive storage, since many XY coordinates point must be stored.

In the same situation, chain-encoded data and skeleton data require roughly equivalent space, generally significantly less than that required by the XY coordinate data. All three representations require approximately the same storage if less exact reconstructions of the figure are acceptable.

If the maps are to be maintained and continuously updated (as in the Duke University study) then the chain-encoded or skeleton representation would be best. For long term storage of data, minimum length files are most desirable. If the files are to be maintained for a relatively short period of time, then less emphasis need be put on storage considerations.

7.4 GRID LIMITATIONS

The graphical display has three types of manual interrupt, that is, three means by which a user can communicate with the system. These are:

(1) Light Pen Interrupts,

(2) Function Key Interrupts,

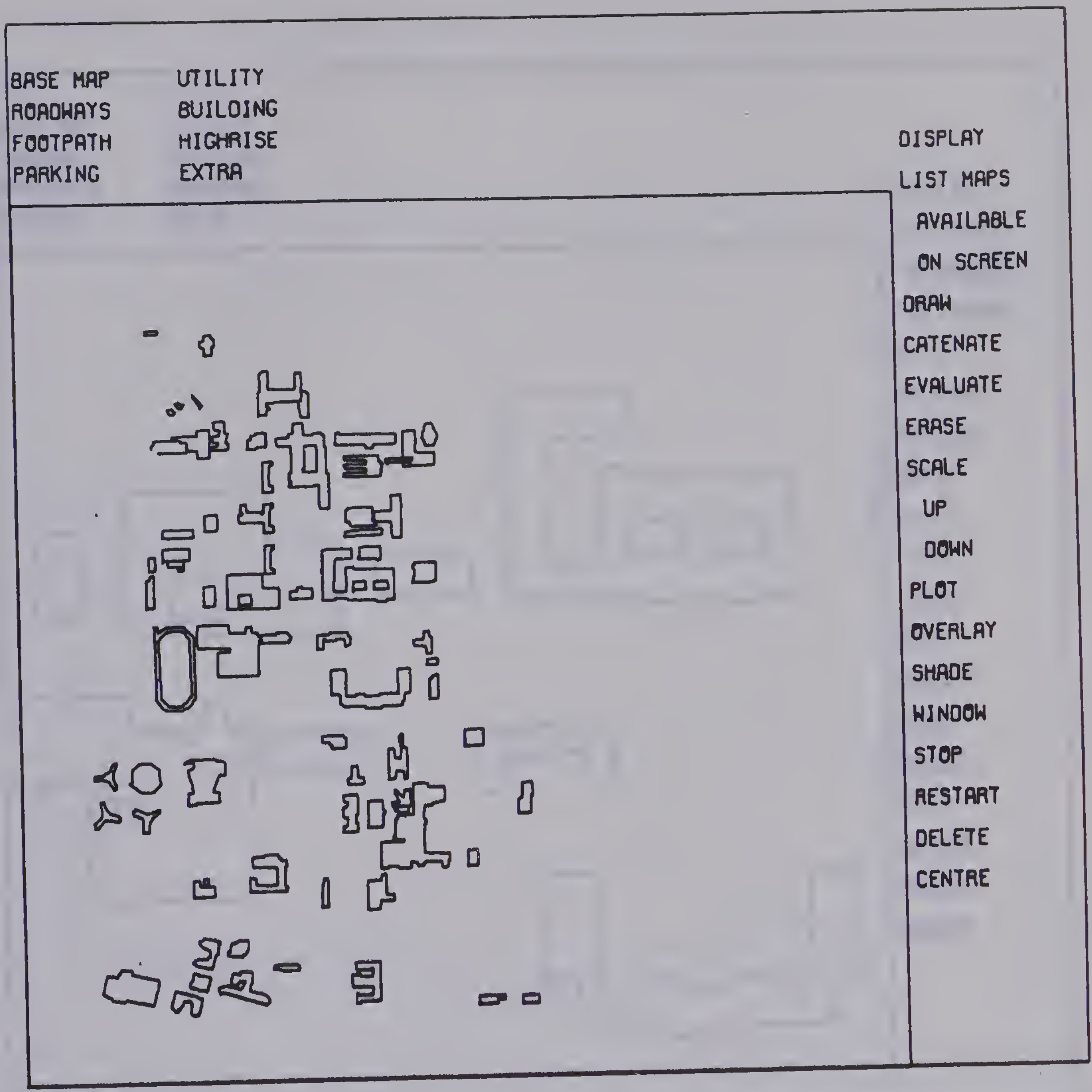
(3) Alphanumeric Key Interrupts.

These actions are common in display systems.

The GRID terminal has (at the present time) 4K 12-bit words of core. The supervisor currently in use occupies 1.5K of this space. Thus the amount of storage available for displaying items on the screen is severely limited. This limitations becomes critical when shaded regions are to be displayed.

The planner has maps of shaded regions. It is impossible to display more than one map at a time (if that) under these conditions if the shading must also be included. To counteract this limitation the map sections displayed on the screen are only the outlines of the regions. In general this is sufficient when the scale of presentation is small. For example, figure 7.2 shows outlines of some of the present buildings on campus. Figure 7.3 shows an enlarged portion of figure 7.2; shading is still not required.

However, an outline is unsatisfactory when the scale of presentation is large. The <SHADE> command has been included for this reason. Only that portion of the map <MAPNAME> which is currently on the screen will be shaded. Figure 7.4 shows the result of applying the shading function to an enlarged portion of the map in figure 7.3. In this way the use of core in the GRID is kept to a minimum without seriously affecting the implementation of the Sieve Process.



PLØT 1 13 JULY 1970

Figure 7.2 Campus Map

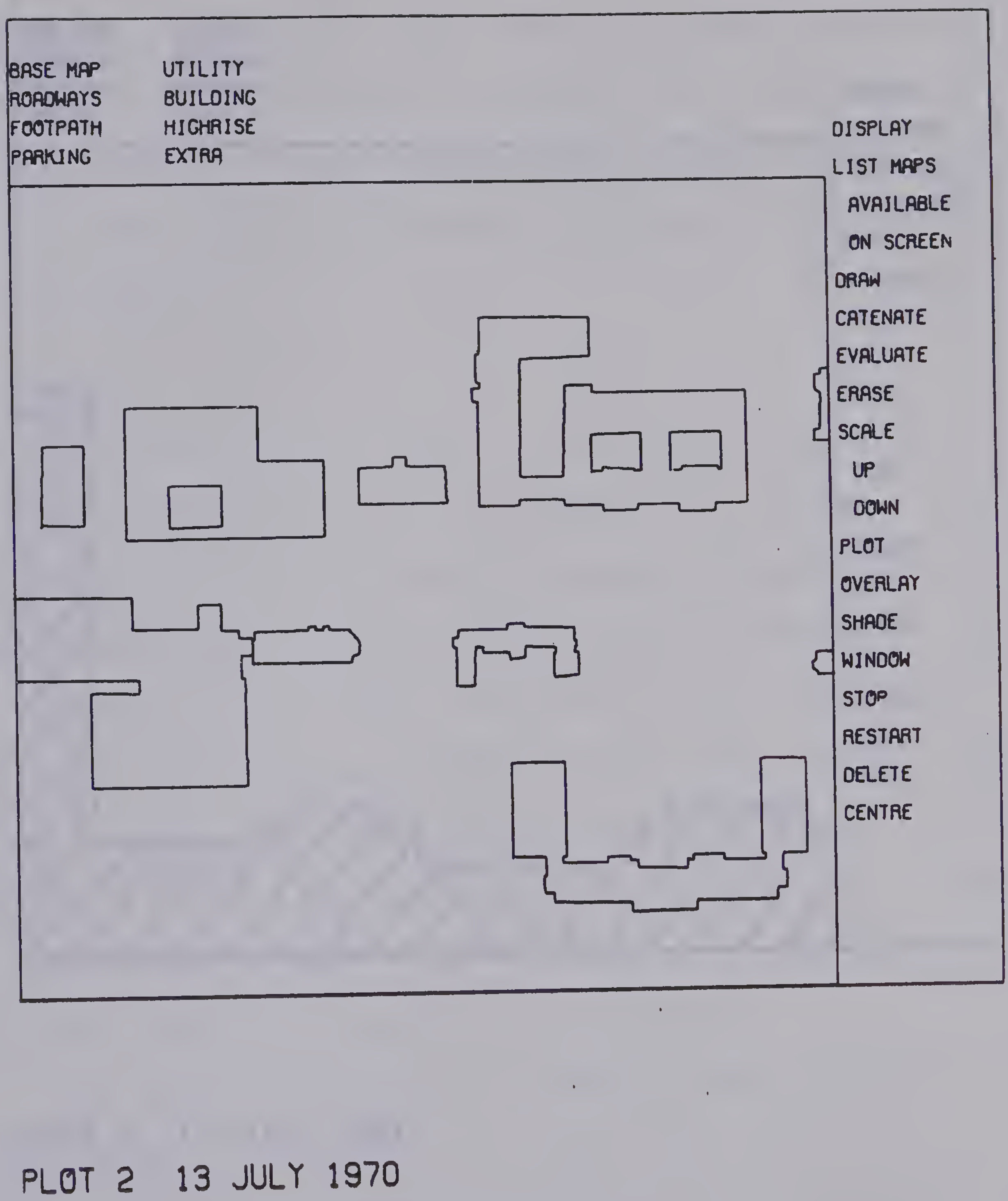


Figure 7.3 Enlarged Campus Map

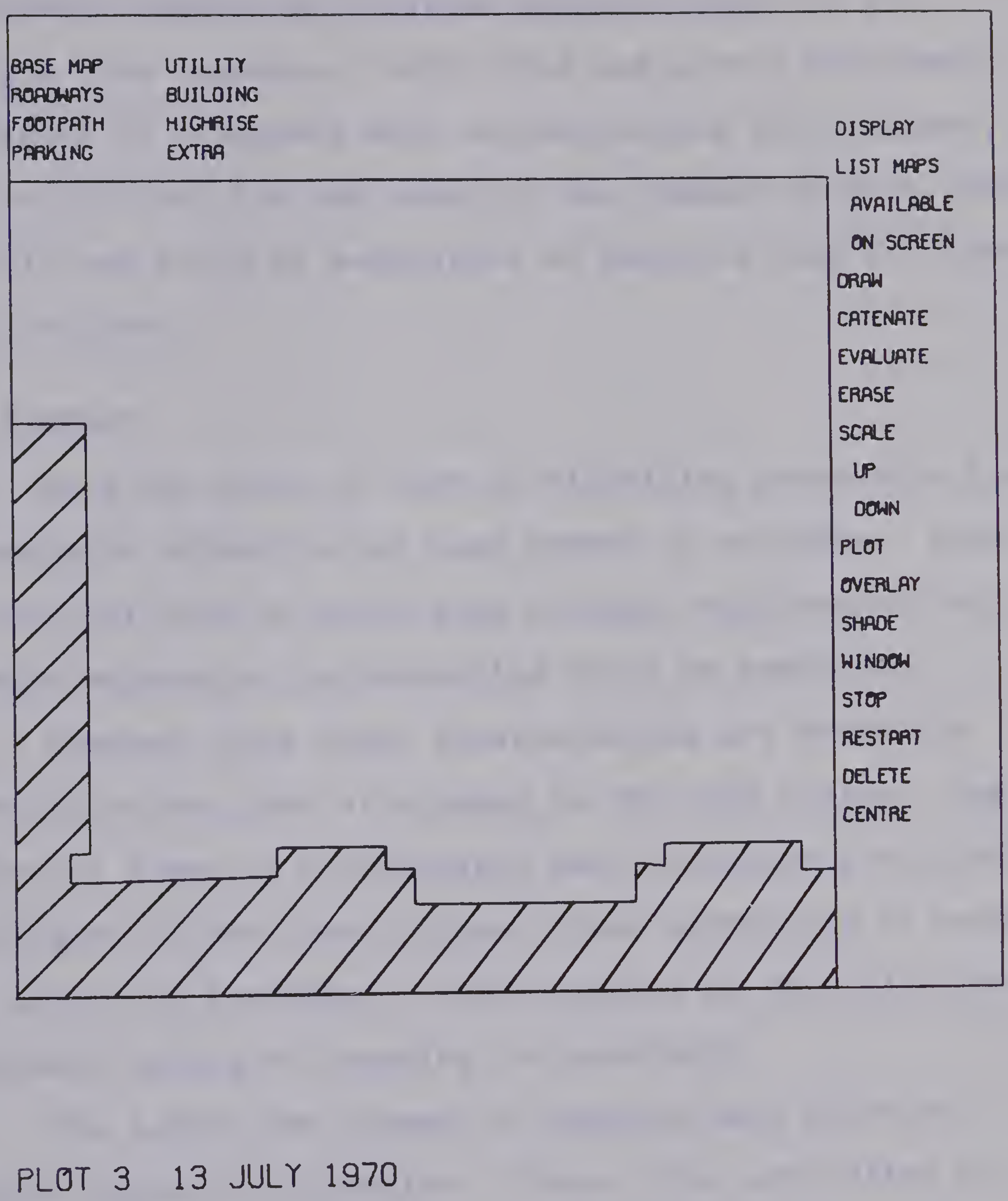


Figure 7.4 Shaded Campus Map

A second method of conserving storage in GRID is to approximate regions by polygons composed of 10 to 15 straight line segments. While this may give a relatively low degree of exactness when reconstructing the drawings, it is sufficient for the needs of the planner in most cases. The full map could be maintained on magnetic tape for more exact analysis.

7.5 SUMMARY

From the point of view of minimizing processing time the skeleton method is the best method of encoding. From the point of view of minimizing storage requirements the skeleton method or chain-encoding would be preferred.

However, both these considerations are presently overruled by the lack of storage in the GRID system. The regions of a map to be displayed must necessarily be limited to polygons of less than fifteen sides especially if several maps are to be overlayed. With polygons of this size the XY coordinate method of encoding is acceptable.

The author has chosen to maintain data files of skeleton encoded information. These files are edited to obtain the polygons required and separate XY coordinate files are maintained. If the user's request changes the screen image, the less exact data is used. If the user's request is for plotted output, the skeleton encoded data is used to obtain a finer drawing than that presented on the CRT. In this manner, at the expense of extra storage in the 360/67, the best merits of both methods are exploited.

CHAPTER VIII

EXTENSIONS TO THE PLANNING SYSTEM

8.1 INTRODUCTION

An interactive system for the Sieve Process has been implemented by the author to the extent described in Chapter 4. It appears that what has been done so far may be used as a partial basis for more extensive systems. For example, the design of the building might be included. Regional and town planning could also be feasible using a similar, but much expanded, system.

Before developing any more elaborate system, however, at least two important items must be considered in greater detail. These are:

(1) Methods of organizing data in an overall form which would facilitate data management and readily allow expansion of the system.

(2) Ways in which the goals of a project might be formulated both for input to the system, and for possible revision by computer program.

8.2 THE SYSTEM AS A BASIS FOR A PLANNING TOOL

The Sieve Process is intended for use in land-site selection problems. However, the land-site selection problem is generally only one part of a larger problem, that of designing and constructing a new building.

When the planner has decided on the best alternative for the land-site, the computer contains information relevant to remaining tasks in the design process. The Duke University study, discussed in Section 2, shows one potential path for development of the system in the university environment. The present implementation consists of maintaining site data and using it to analyse the various possible locations for a particular facility. Figure 2.1 shows that this is within the central part of an overall scheme in which not only is the location decided, but various designs for the building are studied. By extending the current implementation, much of the process could be included. Aguilar (1967b, 1968), Alexander (1964) and Lynch (1962), describe various treatments of the problem of building design, or of design in general, for which the data obtained by using the Sieve Process may be applied. Bernholtz and Bierstone (1966) give a detailed example of one design process to select the optimal design of a house based on Alexander's method of "hierarchical decomposition". This method parallels that of the Sieve Process where the interaction of requirements and operating systems is the dominant factor in approaching the solution.

The Sieve Process has been considered, in this thesis, only in the university environment. This does not mean to say that it is not usable in other areas of planning. Regional and town planning as well as planning of urban renewal schemes might be successfully implemented along

similar lines. Many current schemes under consideration by cities such as Edmonton and Vancouver show that planners make extensive use of the map form in their considerations. In many regions of the country, vast data banks of information concerning urban renewal plans are currently held on file. A substantially extended version of the interactive system discussed by the author could be an ideal way for the planner to reference or modify this information on file.

8.3 COMPUTER MODEL OF THE PLANNING ENVIRONMENT

Inherent in the complex system extensions described above is the notion that the computer would maintain a clear and precise description (or model) of the campus (or other area under consideration) as it relates to the planning task. The current system has no formal basis for data management. Orderly expansion of the system would be much easier if a well defined, and reasonably formal data structure were outlined. One example of such a data structure applied to map data has been developed by Cook (1967). Currently, studies are proceeding at the University of Alberta towards the implementation of a data structure system for use with the GRID.

8.4 INTERACTIVE USE OF GOALS, THEIR FORMULATION AND REVISION

The current system depends entirely on the planner's knowledge of the goals. All processes are performed by the system without any explicit information concerning these

objectives. It seems worthwhile to investigate the possibility of specifying the goals to the system. A result would be that the system would be able to "prod" the planner in a direction indicated by the goals.

Some method of formulating goals for input to the system would then be required. Since goals may change during a session at the CRT means of revising the goals by computer should also be considered. The formulation of goals for input to the system might be very difficult in some cases. Consider, for example, the Student Union Building example outlined in Section 3.2.3. One goal is: "be accessible for service". If a roadway is the access route, then there are restrictions on the road type dependent on the type of vehicular access required. If the access is by means of a parking lot (as is the case at the University of Alberta) then the roadway access to the parking lot must be considered. In fact if the building is only accessible on one or two sides then the ultimate design of the building must be considered before the decision can be made that this goal is fulfilled.

Although the planner may not be seriously hampered by the lack of the facility of computer "prodding" with the current system, any complex planning system would require the computer to maintain and update the planning goals. Thus, we might have two possible paths: "prediction", where the computer prods the user; and "choice", where the user exercises his own judgment or intuition.

8.5 SUMMARY

The present system seems to be a practical first step towards a solution to a wide range of problems using interactive graphics. Prerequisites for the extension of the system include the specification and use of a more formally defined data structure, and a procedure for goal formulation and revision.

CHAPTER IX

CONCLUSIONS

The author has considered the methodology of planning in general and chosen one planning process, the Sieve Process, for implementation using interactive graphic techniques. During implementation, many problems have been brought to light that are common to most map storage and retrieval applications.

The use of digital computers requires digital encoding of maps. In this paper three methods of encoding data were considered. It was found that, since no one method is best overall, the functions required for any application should be considered seriously before the choice of encoding method is made.

In most map storage and retrieval applications, the CRT may be used as if it were a window over a large map, which we can conceive as being drawn on a very large imaginary surface. With appropriate software the user can view any portion of the map to any degree of detail. The method varies according to the mode of digital encoding selected. Though one of the most common problems in graphics, "windowing" has received little study. The author has developed windowing algorithms for use with chain-encoded data or XY coordinate pairs.

The limited system prepared has shown that such procedures as the Sieve Process may be successfully implemented using interactive graphics techniques, and that

the system offers a basis for a more extensive study of the planning process.

Planners now appear to be realizing the potential of the computer as a tool. The main demand is for interaction between the planner and the computer, so that the planner may maintain control over intermediate results and modify them if he desires. Planners now realize that a systematic method must be utilized in order to consider all the important alternatives in other than a haphazard manner. Systematization of planning methods will help clarify the issues for developing computer methods. It is now up to the computing scientists to take up the challenge.

Systems such as the one the author has developed are useful for the present, only at an elementary stage. Much research is needed before their use could become widespread. One important area to be considered is that of data structures. Gray (1967) states that: "The aim of Computer Aided Design is to create in the computer a model of the design problem. .... This model may now be tested against the specification and will generally be modified interactively until the design goal is achieved".

In order to create a model of, for example, the resources of the university campus, a formal method of maintaining and updating data must be decided upon, rather than the informal manner used in the current system. This requirement means that data structure systems must be investigated, with a view to selection or creation of a

system suitable for the planning process, or at least for that section of the process which uses the map form as a base.

At the same time general map storage and retrieval systems should be studied in greater detail in order to discern and perhaps correct some of the difficulties encountered using presently known encoding methods.

The problems encountered on implementing the system were found to be general in nature. The solutions obtained are applicable to a much wider range of applications. Thus there is a great possibility of overlapping and redundant research unless the problems considered and the solutions (if any) obtained are documented. The fact that the problems are common to many areas only reinforces this need. In this way a solid base may be built for the field of interactive graphics.

BIBLIOGRAPHY

Aguilar, R.J., 1967a. An Application of Dynamic Programming to Structural Optimization, Division of Engineering Research, Bulletin 91, Louisiana State University, Baton Rouge.

Aguilar, R.J., 1967b. The Mathematical Formulation and Optimization of Architectural and Planning Functions, Division of Engineering Research, Bulletin 93, Louisiana State University, Baton Rouge.

Aguilar, R.J., 1969. Optimality in Facility Location, presented at joint meeting of Canadian Institute of Architects and University of Alberta Extension Division, Edmonton.

Aguilar, R.J. and Hand, J.E., 1968. "A Generalized Linear Model for Optimization of Architectural Planning", Proc. AFIPS, Vol. 32, pp. 81-88.

Alexander, C., 1964. Notes on the Synthesis of Form, Harvard University Press, Cambridge.

Berkeley, E.P., 1968. "Computers for Design and a Design for the Computer", The Architectural Forum, Vol. 128, No. 2, pp. 60-65.

Bernholtz, A. and Bierstone, E., 1966. "Computer-Augmented Design", Design Quarterly 66/67, Walker Art Center, Minneapolis, pp. 41-51.

Blum, H., 1964. "A Transformation for Extracting New Descriptors of Shape", Symp. on Models for the Perception of Speech and Visual Form, MIT Press, Cambridge, pp. 362-379.

Catanese, A.J. and Steiss, P.W., 1968. "Systemic Planning", Journal of the Town Planning Institute, Vol. 54, No. 4, pp. 172-176.

Cook, B.G., 1967. "A Computer Representation of Plane Region Boundaries", The Australian Computer Journal, Vol. 1, No. 1, pp. 44-50.

Cotton, I.W. and Greatorex, Jr. F.S., 1968. "Data Structures and Techniques for Remote Computer Graphics", Proc. AFIPS, Vol. 33, pp. 533-544.

Dror, Y., 1961. Introduction to a General Theory of Planning, Institute of Social Studies, The Hague.

Duke University, 1967. Computer Aided Campus Planning for Colleges and Universities, Interim Report, Duke University.

Eastman, C.M., 1970. "Representations for Space Planning", C.A.C.M., Vol. 13, No. 4, pp. 242-250.

Fair, G.P., Flowerdew, A.D.J., Munro, W.G. and Rowley, D., 1966. "Note on the Computer as an aid to the Architect", Computer Journal, Vol. 9, No. 1, pp. 16-20.

Freeman, H., 1961a. "On the Encoding of Arbitrary Geometric Configurations", Transactions of the IRE, Vol. EC-10, No. 2, pp. 260-268.

Freeman, H., 1961b. "Techniques for Digital Computer Analysis of Chain Encoded Arbitrary Plane Curves", Proc. National Electronics Conference, Chicago, Vol. 17, pp. 421-432.

Freeman, H., 1967. "On the Classification of Line-Drawing Data", Symp. on Models for the Perception of Speech and Visual Form, MIT Press, Boston, pp. 408-412.

Gray, J.C., 1967. "Compound Data Structure for Computer Aided Design; a Survey", Proc. 22nd ACM Nat. Conf., pp. 355-365.

Herzog, B., 1968. "Computer Graphics for Designers", Emerging Concepts in Computer Graphics, D. Secrest and J. Nievergelt (editors), W.A. Benjamin Inc., New York, pp. 189-230.

Huen, W.H., 1969. A Graphical Display Subroutine Package, (M.Sc. Thesis), Dept. of Computing Science, University of Alberta, Edmonton.

Jacks, E.L., 1966. "Design Augmented by Computer", Design Quarterly 66/67, Walker Art Center, Minneapolis, pp. 25-30.

Jacobsen, J.B., May, K.L., Huen, W.H. and Penny, J.P., 1970. Computer Graphics for the Fortran Programmer, Dept. of Computing Science, University of Alberta, Edmonton.

Lichfield, N., 1969. "Goals in Planning", Journal of the Town Planning Institute, Vol. 55, No. 1, pp. 21-27.

Loomis, R.G., 1965. "Boundary Networks", C.A.C.M., Vol. 8, No. 1, pp. 44-48.

Lynch, K., 1962. Site Planning, MIT Press, Cambridge.

Manheim, M.L., 1966. "Problem Solving Processes in Planning and Design", Design Quarterly 66/67, Walker Art Center, Minneapolis, pp. 31-40.

Mezei, L., 1967. See drawings in Computers and Automation, Vol. 16, No. 8, p. 9.

Mezei, L., 1968a. See drawings in Computers and Automation, Vol. 17, No. 8, pp. 12-13.

Mezei, L., 1968b. "SPARTA, a Procedure Oriented Programming Language for the Manipulation of Arbitrary Line Drawings", IFIP Conf. Proc., pp. C96-C102.

Montanari, U., 1968. "A Method for Obtaining Skeletons Using a Quasi-Euclidean Distance", J.A.C.M., Vol. 15, No. 4, pp. 600-624.

Montanari, U., 1969. "Continuous Skeletons From Digitized Images", J.A.C.M., Vol. 16, No. 4, pp. 534-549.

Montanari, U., 1970a. "A Note on Minimal Length Polygonal Approximation to a Digitized Contour", C.A.C.M., Vol. 13, No. 1, pp. 41-47.

Montanari, U., 1970b. "On Limit Properties in Digitization Schemes", J.A.C.M., Vol. 17, No. 2, pp. 348-360.

Morse, S.P., 1968a. "A Mathematical Model for the Analysis of Contour-line Data", J.A.C.M., Vol. 15, No. 2, pp. 205-220.

Morse, S.P., 1968b. "Computer Storage of Contour Map Data", Proc. 23rd ACM National Conf., pp. 45-51.

Newman, W.M., 1966. "Experimental Program for Architectural Design, Computer Journal, Vol. 9, No. 1, pp. 21-26.

Parker, D.B., 1967. Solving Design Problems in Graphical Dialogue, Programming Technical Report TER-01, C.D.C., Pala Alto.

Pfaltz, J.L. and Rosenfeld, A., 1967. "Computer Representation of Planar Regions by their Skeletons", C.A.C.M., Vol. 10, No. 2, pp. 119-125.

Pfaltz, J.L., Snively, J.W. and Rosenfeld, A., 1968. "Local and Global Picture Processing by Computer", Pictorial Pattern Recognition, Thompson Book Co., pp. 353-371.

Rosenfeld, A., 1969. "Picture Processing by Computer", Computing Surveys, Vol. 1, No. 3, pp. 146-176.

Rosenfeld, A., 1970. "Connectivity in Digital Pictures", J.A.C.M., Vol. 17, No. 1, pp. 146-160.

Rosenfeld, A. and Pfaltz, J.L., 1966. "Sequential Operations in Digital Picture Processing", J.A.C.M., Vol. 13, No. 4, pp. 471-494.

Ross, D.T., 1967. "The AED Approach to Generalized Computer-aided Design", Proc. 22nd ACM National Conf., pp. 367-385.

Rubenstein, H.M., 1969. A Guide to Site and Environmental Planning, John Wiley and Sons, New York.

Schofer, R.E. and Goodyear, F.F., 1967. "Electronic Computer Applications in Urban Transportation Planning", Proc. 22nd ACM National Conf., pp. 247-253.

Spreiregen, P.D., 1965. Urban Design: The Architecture of Towns and Cities, McGraw-Hill Book Co., New York.

Sutherland, I.E. and Sproull, R.F., 1968. "A Clipping Divider", Proc. AFIPS, Vol. 33, pp. 765-776.

University of Alberta, 1967. The University of Alberta North Garneau Development: Report 1, Bittorf Pinckston Planning Consultants.

University of Alberta, 1968a. The University of Alberta North Garneau Development: Report 2, Bittorf Pinckston Planning Consultants.

University of Alberta, 1968b. The University of Alberta North Garneau Development; Interim Report: Planning Unit, Bittorf Pinckston Planning Consultants.

University of Alberta, 1968c. The University of Alberta North Garneau Development; Interim Report: Space Hierarchy, Bittorf Pinckston Planning Consultants.

Wurman, R.S. and Killinger, S.W., 1967. "Visual Information Systems", Architecture Canada, Vol. 44, No. 3, pp. 37-62.